Benutzer-Leitfaden

AWS CodeDeploy



API-Version 2014-10-06

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

AWS CodeDeploy: Benutzer-Leitfaden

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

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist CodeDeploy? ... 1
Vorteile von AWS CodeDeploy ... 2
Überblick über CodeDeploy Rechenplattformen ... 3
Überblick über die CodeDeploy Bereitstellungstypen ... 11
Überblick über eine direkte Bereitstellung ... 13
Überblick über eine Bereitstellung blue/green ... 14
Bitte geben Sie uns Feedback ... 19
Primäre Komponenten ... 19
Anwendung ... 20
Plattform für die Datenverarbeitung ... 20
Bereitstellungskonfiguration ... 21
Bereitstellungsgruppe ... 22
Bereitstellungstyp ... 22
Bereitstellungstyp ... 23
Revision ... 24
Service – -Rolle ... 24
Zielrevision ... 24
Andere Komponenten ... 25
Bereitstellungen ... 25
Bereitstellungen auf einem AWS Lambda Compute-Plattform ... 25
Bereitstellung auf einer Amazon-ECS-Datenverarbeitungsplattform ... 29
Bereitstellungen auf einer Rechenplattform EC2/On-Premises ... 42
Dateien mit Anwendungsspezifikationen ... 49
AppSpec Dateien auf einer Amazon ECS Compute Platform ... 50
AppSpec Dateien auf einem AWS Lambda Plattform für die Datenverarbeitung ... 50
AppSpec Dateien auf einer EC2/on-premises Rechenplattform ... 50
Wie der CodeDeploy Agent die AppSpec Datei verwendet ... 51
Erste Schritte ... 52
Schritt 1: Einrichtung ... 52
Melden Sie sich an für eine AWS-Konto ... 52
Schritt 2: Erstellen Sie eine Servicerolle ... 53
Erstellen Sie eine Servicerolle (Konsole) ... 55
Erstellen Sie eine Servicerolle (CLI) ... 58
Rufen Sie die Servicerolle ARN ab (Konsole) ... 61

Rufen Sie die Servicerolle ARN (CLI) ab ... 61
Schritt 3: Beschränken Sie die CodeDeploy Benutzerberechtigungen ... 62
Schritt 4: Erstellen Sie ein IAM-Instance-Profil ... 66
Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances (CLI) ... 67
Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances (Konsole) ... 71
Produkt- und Service-Integrationen ... 75
Integration mit anderen AWS Diensten ... 75
Amazon EC2 Auto Scaling ... 84
Elastic Load Balancing ... 93
Integration mit Produkten und Dienstleistungen von Partnern ... 98
GitHub ... 106
Integrationsbeispiele der Community ... 110
Blog-Posts ... 110
Lernprogramme ... 112
Tutorial: Auf einer WordPress Nicht-Windows-Instance bereitstellen ... 112
Schritt 1: Starten einer Amazon-EC2-Instance ... 113
Schritt 2: Konfigurieren Sie Ihren Quellinhalt ... 116
Schritt 3: Laden Sie Ihre Bewerbung auf Amazon S3 hoch ... 122
Schritt 4: Stellen Sie Ihre Anwendung bereit ... 127
Schritt 5: Aktualisieren Sie Ihre Anwendung und stellen Sie sie erneut bereit ... 134
Schritt 6: Bereinigen ... 139
Tutorial: Stellen Sie eine Hello World-Anwendung auf einer Windows Server-Instanz bereit ... 143
Schritt 1: Starten einer Amazon-EC2-Instance ... 144
Schritt 2: Konfigurieren Sie Ihren Quellinhalt ... 146
Schritt 3: Laden Sie Ihre Bewerbung auf Amazon S3 hoch ... 149
Schritt 4: Stellen Sie Ihre Anwendung bereit ... 155
Schritt 5: Aktualisieren Sie Ihre Anwendung und stellen Sie sie erneut bereit ... 160
Schritt 6: Bereinigen ... 164
Tutorial: Stellen Sie eine Anwendung auf einer lokalen Instanz bereit ... 167
Voraussetzungen ... 168
Schritt 1: Konfigurieren Sie die lokale Instanz ... 169
Schritt 2: Erstellen Sie eine Version der Beispielanwendung ... 169
Schritt 3: Bündeln Sie Ihre Anwendungsrevision und laden Sie sie auf Amazon S3 hoch ... 174
Schritt 4: Stellen Sie die Revision Ihrer Anwendung bereit ... 175
Schritt 5: Überprüfen Sie Ihre Bereitstellung ... 175
Schritt 6: Bereinigen von Ressourcen ... 175

Tutorial: In einer Auto Scaling Scaling-Gruppe bereitstellen 178
Voraussetzungen 178
Schritt 1: Auto Scaling Scaling-Gruppe erstellen und konfigurieren 179
Schritt 2: Bereitstellen der Anwendung in der Auto Scaling Scaling-Gruppe 186
Schritt 3: Überprüfen Sie Ihre Ergebnisse 196
Schritt 4: Erhöhen Sie die Anzahl der Amazon EC2 EC2-Instances in der Auto Scaling Scaling-Gruppe 198
Schritt 5: Überprüfen Sie Ihre Ergebnisse erneut 199
Schritt 6: Bereinigen 202
Tutorial: Stellen Sie eine Anwendung bereit von GitHub 204
Voraussetzungen 205
Schritt 1: Richten Sie ein GitHub Konto ein 205
Schritt 2: Erstellen Sie ein GitHub Repository 206
Schritt 3: Laden Sie eine Beispielanwendung in Ihr GitHub Repository hoch 208
Schritt 4: Stellen Sie eine Instanz bereit 213
Schritt 5: Erstellen Sie eine Anwendung und eine Bereitstellungsgruppe 214
Schritt 6: Stellen Sie die Anwendung auf der Instanz bereit 216
Schritt 7: Überwachen und verifizieren Sie die Bereitstellung 221
Schritt 8: Bereinigen 222
Tutorial: Bereitstellen einer Anwendung in Amazon ECS 224
Voraussetzungen 226
Schritt 1: Aktualisieren Sie Ihre Amazon ECS-Anwendung 227
Schritt 2: Erstellen Sie die Datei AppSpec 228
Schritt 3: Verwenden Sie die CodeDeploy Konsole, um Ihre Anwendung bereitzustellen 230
Schritt 4: Bereinigen 234
Tutorial: Bereitstellen eines Amazon ECS-Service mit einem Validierungstest 235
Voraussetzungen 238
Schritt 1: Erstellen Sie einen Test-Listener 238
Schritt 2: Aktualisieren Sie Ihre Amazon ECS-Anwendung 239
Schritt 3: Erstellen Sie eine Lifecycle-Hook-Lambda-Funktion 239
Schritt 4: Aktualisieren Sie Ihre AppSpec Datei 242
Schritt 5: Verwenden Sie die CodeDeploy Konsole, um Ihren Amazon ECS-Service bereitzustellen 243
Schritt 6: Zeigen Sie die Ausgabe Ihrer Lambda-Hook-Funktion in Logs an CloudWatch 245
Schritt 7: Bereinigen 246
Tutorial: Eine Lambda-Funktion mit AWS SAM bereitstellen 247

Voraussetzungen ... 248
Schritt 1: Richten Sie Ihre Infrastruktur ein ... 248
Schritt 2: Aktualisieren Sie die Lambda-Funktion ... 264
Schritt 3: Stellen Sie die aktualisierte Lambda-Funktion bereit ... 267
Schritt 4: Zeigen Sie Ihre Bereitstellungsergebnisse an ... 269
Schritt 5: Bereinigen ... 272
Mit dem CodeDeploy Agenten zusammenarbeiten ... 274
Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy ... 275
Unterstützte Amazon EC2 AMI-Betriebssysteme ... 275
Unterstützte lokale Betriebssysteme ... 275
Kommunikationsprotokoll und Port für den CodeDeploy Agenten ... 276
Versionsverlauf des Agenten CodeDeploy ... 277
Den CodeDeploy Prozess verwalten ... 297
Revision der Anwendung und Bereinigung der Protokolldateien ... 298
Vom Agenten installierte Dateien CodeDeploy ... 299
Verwaltung des CodeDeploy Agentenbetriebs ... 303
Stellen Sie sicher, dass der CodeDeploy Agent läuft ... 303
Ermitteln Sie die Version des Agenten CodeDeploy ... 306
Installieren Sie den CodeDeploy Agenten ... 309
Aktualisieren Sie den CodeDeploy Agenten ... 323
Deinstallieren Sie den CodeDeploy Agenten ... 326
CodeDeploy Agentenprotokolle senden an CloudWatch ... 327
Arbeiten mit Instances ... 333
Vergleich von Amazon EC2 EC2-Instances mit lokalen Instances ... 333
Instanzaufgaben für CodeDeploy ... 335
Markieren von Instanzen für Bereitstellungen CodeDeploy ... 337
Beispiel 1: Einzelne Tag-Gruppe, einzelnes Tag ... 338
Beispiel 2: Einzelne Tag-Gruppe, mehrere Tags ... 339
Beispiel 3: Mehrere Tag-Gruppen, einzelne Tags ... 341
Beispiel 4: Mehrere Tag-Gruppen, mehrere Tags ... 343
Arbeiten mit Amazon EC2 EC2-Instances ... 347
Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy ... 347
Eine Amazon EC2 EC2-Instance erstellen (CloudFormation Vorlage) ... 354
Eine Amazon EC2 EC2-Instance konfigurieren ... 365
Arbeiten mit On-Premises-Instances ... 370
Voraussetzungen für die Konfiguration einer lokalen Instanz ... 371

On-Premises-Instance registrieren ... 373
Verwaltung des Betriebs von lokalen Instanzen ... 411
Anzeigen von Instance-Details ... 419
Instanzdetails anzeigen (Konsole) ... 420
Instanzdetails anzeigen (CLI) ... 421
Gesundheit der Instanz ... 421
Gesundheitsstatus ... 422
Ungefähr die Mindestanzahl fehlerfreier Instanzen ... 423
Ungefähr die Mindestanzahl fehlerfreier Instanzen pro Availability Zone ... 428
Arbeiten mit Bereitstellungskonfigurationen ... 430
Bereitstellungskonfigurationen auf einer EC2/on-premises Rechenplattform ... 430
Vordefinierte Bereitstellungskonfigurationen ... 430
Bereitstellungskonfigurationen auf einer Amazon-ECS-Datenverarbeitungsplattform ... 436
Vordefinierte Bereitstellungskonfigurationen für Amazon ECS ... 437
Bereitstellungskonfigurationen für CloudFormation blue/green Bereitstellungen (Amazon ECS) ... 438
Bereitstellungskonfigurationen auf einem AWS Lambda Plattform für die Datenverarbeitung ... 438
Vordefinierte Bereitstellungskonfigurationen für Lambda ... 439
Topics ... 440
Erstellen Sie eine Bereitstellungskonfiguration ... 440
Erstellen Sie eine Bereitstellungskonfiguration (Konsole) ... 441
Erstellen Sie eine Bereitstellungskonfiguration (AWS CLI) ... 443
Details zur Bereitstellungskonfiguration anzeigen ... 445
Details zur Bereitstellungskonfiguration anzeigen (Konsole) ... 445
Bereitstellungskonfiguration anzeigen (CLI) ... 446
Löschen Sie eine Bereitstellungskonfiguration ... 446
Arbeiten mit -Anwendungen ... 447
Erstellen einer Anwendung ... 448
Erstellen Sie eine Anwendung für eine direkte Bereitstellung (Konsole) ... 450
Erstellen Sie eine Anwendung für eine blue/green Bereitstellung (Konsole) ... 454
Eine Anwendung für eine Amazon ECS-Servicebereitstellung erstellen (Konsole) ... 459
Erstellen Sie eine Anwendung für ein AWS Lambda Bereitstellung von Funktionen (Konsole) ... 462
Anwendung erstellen (CLI) ... 464
Anwendungsdetails anzeigen ... 464
Anwendungsdetails anzeigen (Konsole) ... 465

Anwendungsdetails anzeigen (CLI) ... 465
Erstellen einer Benachrichtigungsregel ... 465
Benennen Sie eine Anwendung um ... 469
Löschen einer Anwendung ... 469
Löschen Sie eine Anwendung (Konsole) ... 470
Löschen Sie eine Anwendung (AWS CLI) ... 470
Arbeiten mit Bereitstellungsgruppen ... 471
Bereitstellungsgruppen in Bereitstellungen auf Amazon-ECS-Datenverarbeitungsplattformen ... 471
Bereitstellungsgruppen in AWS Lambda Bereitstellungen von Rechenplattformen ... 471
Bereitstellungsgruppen in EC2/On-Premises Compute Platform-Bereitstellungen ... 471
Topics ... 472
Erstellen Sie eine Bereitstellungsgruppe ... 473
Erstellen Sie eine Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole) ... 473
Erstellen Sie eine Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung (Konsole) ... 478
Eine Bereitstellungsgruppe für eine Amazon ECS-Bereitstellung erstellen (Konsole) ... 485
Richten Sie in Elastic Load Balancing einen Load Balancer für CodeDeploy Amazon EC2 EC2-Bereitstellungen ein ... 487
Einen Load Balancer, Zielgruppen und Listener für CodeDeploy Amazon ECS-Bereitstellungen einrichten ... 488
Erstellen Sie eine Bereitstellungsgruppe (CLI) ... 494
Details zur Bereitstellungsgruppe anzeigen ... 495
Details zur Bereitstellungsgruppe anzeigen (Konsole) ... 496
Details zur Bereitstellungsgruppe anzeigen (CLI) ... 496
Ändern Sie die Einstellungen der Bereitstellungsgruppe ... 497
Einstellungen der Bereitstellungsgruppe ändern (Konsole) ... 497
Einstellungen für Bereitstellungsgruppen ändern (CLI) ... 498
Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe ... 500
Löschen Sie eine Bereitstellungsgruppe ... 504
Löschen Sie eine Bereitstellungsgruppe (Konsole) ... 504
Eine Bereitstellungsgruppe löschen (CLI) ... 505
Arbeiten mit Anwendungsrevisionen ... 506
Planen Sie eine Revision ... 506
Fügen Sie eine AppSpec Datei hinzu ... 507
Fügen Sie eine AppSpec Datei für eine Amazon ECS-Bereitstellung hinzu ... 507
Fügen Sie eine AppSpec Datei hinzu für AWS Lambda-Bereitstellung ... 511

Fügen Sie eine AppSpec Datei für eine EC2/On-Premises Bereitstellung hinzu 513
Wählen Sie einen Repository-Typ 517
Eine Revision pushen 520
Pushen Sie eine Revision mit dem AWS CLI 522
Details zur Anwendungsversion anzeigen 525
Details zur Anwendungsversion anzeigen (Konsole) 525
Revisionsdetails der Anwendung anzeigen (CLI) 526
Registrieren Sie eine Anwendungsrevision 527
Registrieren Sie eine Revision in Amazon S3 mit CodeDeploy (CLI) 527
Eine Revision registrieren GitHub mit CodeDeploy (CLI) 529
Arbeiten mit Bereitstellungen 530
Eine Bereitstellung auswählen 531
Voraussetzungen für die Bereitstellung 532
Erstellen Sie eine Amazon ECS Compute Platform-Bereitstellung (Konsole) 535
Erstellen Sie ein AWS Bereitstellung der Lambda Compute Platform (Konsole) 537
Erstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (Konsole) 540
Erstellen Sie eine Amazon ECS Compute Platform-Bereitstellung (CLI) 545
Erstellen Sie ein AWS Bereitstellung der Lambda Compute Platform (CLI) 546
Erstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (CLI) 548
Erstellen Sie eine Amazon blue/green ECS-Bereitstellung über CloudFormation 552
Bereitstellungsdetails anzeigen 556
Bereitstellungsdetails anzeigen (Konsole) 557
Bereitstellungsdetails anzeigen (CLI) 558
Bereitstellungsprotokolldaten anzeigen 559
Protokolldateidaten in der CloudWatch Amazon-Konsole anzeigen 559
Logdateien auf einer Instanz anzeigen 559
Stoppen Sie eine Bereitstellung 563
Stoppen einer Bereitstellung (Konsole) 564
Stoppen einer Bereitstellung (CLI) 565
Eine Bereitstellung erneut bereitstellen und rückgängig machen 565
Automatische Rollbacks 565
Manuelle Rollbacks 566
Arbeitsablauf für Rollback und erneutes Deployment 566
Rollback-Verhalten bei vorhandenem Inhalt 568
Stellen Sie eine Anwendung in einem anderen AWS Konto bereit 572
Schritt 1: Erstellen Sie in einem der Konten einen S3-Bucket 573

Schritt 2: Erteilen Sie Amazon S3 S3-Bucket-Berechtigungen für das IAM-Instance-Profil des Produktionskontos ... 573
Schritt 3: Erstellen Sie Ressourcen und eine kontoübergreifende Rolle im Produktionskonto ... 574
Schritt 4: Laden Sie die Anwendungsrevision in den Amazon S3 S3-Bucket hoch ... 576
Schritt 5: Übernehmen Sie die kontoübergreifende Rolle und stellen Sie Anwendungen bereit ... 576
Validieren Sie ein Bereitstellungspaket auf einem lokalen Computer ... 577
Voraussetzungen ... 577
Erstellen Sie eine lokale Bereitstellung ... 580
Beispiele ... 583
Überwachen von Bereitstellungen ... 586
Automatisierte Tools ... 586
Manuelle Tools ... 588
Überwachung von Bereitstellungen mit Amazon-Tools CloudWatch ... 589
Überwachung von Bereitstellungen mit Alarmen CloudWatch ... 589
Überwachung von Bereitstellungen mit Amazon Events CloudWatch ... 591
Überwachung von Bereitstellungen mit AWS CloudTrail ... 594
CodeDeploy Informationen in CloudTrail ... 594
Grundlegendes zu CodeDeploy Protokolldateieinträgen ... 595
Überwachung von Bereitstellungen mit Amazon SNS SNS-Ereignisbenachrichtigungen ... 597
Gewähren Sie Amazon SNS SNS-Berechtigungen für eine Servicerolle ... 598
Erstellen Sie einen Auslöser für ein CodeDeploy Ereignis ... 599
Bearbeiten Sie einen Auslöser in einer Bereitstellungsgruppe ... 607
Löschen Sie einen Trigger aus einer Bereitstellungsgruppe ... 609
JSON-Datenformate für Trigger ... 610
Sicherheit ... 613
Datenschutz ... 614
Richtlinie für den Datenverkehr zwischen Netzwerken ... 615
Verschlüsselung im Ruhezustand ... 615
Verschlüsselung während der Übertragung ... 616
Verwaltung von Verschlüsselungsschlüsseln ... 616
Identity and Access Management ... 616
Zielgruppe ... 617
Authentifizierung mit Identitäten ... 617
Verwalten des Zugriffs mit Richtlinien ... 618

Wie AWS CodeDeploy funktioniert mit IAM ... 620
AWS verwaltete (vordefinierte) Richtlinien für CodeDeploy ... 625
CodeDeploy Aktualisierungen zu AWS Verwaltete Richtlinien ... 634
Identity-based Beispiele für politische Maßnahmen ... 638
Fehlerbehebung ... 646
CodeDeploy Referenz zu Berechtigungen ... 648
Cross-service verwirrter Abgeordneter, Prävention ... 657
Vorfallreaktion ... 659
Prüfung aller Interaktionen mit CodeDeploy ... 659
Alarmierungs- und Vorfallmanagement ... 660
Compliance-Validierung ... 661
Ausfallsicherheit ... 661
Sicherheit der Infrastruktur ... 662
Referenz ... 663
AppSpec Dateireferenz ... 663
AppSpec Dateien auf einer Amazon ECS-Rechenplattform ... 664
AppSpec Dateien auf einem AWS Lambda Plattform für die Datenverarbeitung ... 664
AppSpec Dateien auf einer EC2/on-premises Rechenplattform ... 664
AppSpec Struktur der Datei ... 665
AppSpec Beispiel für eine Datei ... 713
AppSpec Abstand zwischen den Dateien ... 719
Bestätigen Sie Ihre AppSpec Datei und den Speicherort ... 721
Referenz zur Agentenkonfiguration ... 722
Verwandte Themen ... 734
CloudFormation Vorlagenreferenz ... 734
Verwendung CodeDeploy mit Amazon Virtual Private Cloud ... 738
Verfügbarkeit ... 739
VPC-Endpoints erstellen für CodeDeploy ... 742
Konfigurieren Sie die Agenten- und IAM-Berechtigungen CodeDeploy ... 743
Referenz zum Resource Kit ... 744
Bucket-Namen für das Resource Kit nach Regionen ... 744
Inhalt des Resource Kits ... 746
Zeigt eine Liste der Resource Kit-Dateien an ... 749
Laden Sie die Resource Kit-Dateien herunter ... 750
Kontingente ... 754
Fehlerbehebung ... 762

Allgemeine Probleme bei der Fehlerbehebung 762
Allgemeine Checkliste für die Fehlerbehebung 763
CodeDeploy Bereitstellungsressourcen werden nur in einigen Fällen unterstützt AWS Regionen 765
Die Verfahren in diesem Handbuch stimmen nicht mit der CodeDeploy Konsole überein 765
Erforderliche IAM-Rollen sind nicht verfügbar 766
Die Verwendung einiger Texteditoren zum Erstellen von AppSpec Dateien und Shell-Skripts kann dazu führen, dass Bereitstellungen fehlschlagen 766
Verwenden von Finder in macOS zur Bündelung einer Anwendungsrevision kann zum Fehlschlagen von Bereitstellungen führen 767
Beheben Sie Probleme bei der EC2/On-Premises Bereitstellung 767
CodeDeploy Fehler beim Fehlen der Anmeldeinformationen für das Plugin CommandPoller 768
Die Bereitstellung schlägt fehl und es wird die Meldung „Die Validierung der mit PKCS7 signierten Nachricht ist fehlgeschlagen“ angezeigt 769
Bereitstellung oder erneute Bereitstellung derselben Dateien in denselben Instance-Standorten schlägt fehl mit dem Fehler „The deployment failed because a specified file already exists at this location“ 769
Lange Dateipfade führen zu der Fehlermeldung „Keine solche Datei oder kein solches Verzeichnis“ 772
Long-running Prozesse können dazu führen, dass Bereitstellungen fehlschlagen 773
Behebung eines fehlgeschlagenen AllowTraffic Lebenszyklusereignisses, bei dem kein Fehler in den Bereitstellungsprotokollen gemeldet wurde 775
Behebung eines Fehlers oder eines ApplicationStop BeforeBlockTraffic Ereignisses im Lebenszyklus einer AfterBlockTraffic Bereitstellung 775
Behebung eines fehlgeschlagenen DownloadBundle Deployment-Lifecycle-Ereignisses mit UnknownError: nicht zum Lesen geöffnet 777
Durch Fehlerbehebung bei allen Lebenszyklusereignissen wurden Fehler übersprungen 778
PowerShell Windows-Skripts verwenden standardmäßig nicht die 64-Bit-Version von Windows PowerShell 780
Beheben von Amazon ECS-Bereitstellungsproblemen 781
Beim Warten auf den Satz einer Ersatzaufgabe tritt ein Timeout auf 782
Beim Warten auf die Fortsetzung einer Benachrichtigung tritt ein Timeout auf 782
Die IAM-Rolle verfügt nicht über genügend Berechtigungen 783
Beim Warten auf einen Statusrückruf ist bei der Bereitstellung ein Timeout aufgetreten 783

Die Bereitstellung ist fehlgeschlagen, weil eine oder mehrere der Funktionen zur Überprüfung von Lebenszyklusereignissen fehlgeschlagen sind 784
Der ELB konnte aufgrund des folgenden Fehlers nicht aktualisiert werden: Die primäre Taskset-Zielgruppe muss sich hinter dem Listener befinden 785
Meine Bereitstellung schlägt manchmal fehl, wenn Auto Scaling verwendet wird 786
Nur ALB unterstützt das schrittweise Routing des Datenverkehrs. Verwenden Sie stattdessen das AllAtOnce Traffic-Routing, wenn Sie create/update eine Gruppe einrichten 787
Obwohl meine Bereitstellung erfolgreich war, besteht der Ersatz-Tasksatz bei den Elastic Load Balancing-Zustandsprüfungen nicht und meine Anwendung ist ausgefallen 788
Kann ich einer Bereitstellungsgruppe mehrere Load Balancer zuordnen? 789
Kann ich CodeDeploy blue/green Bereitstellungen ohne einen Load Balancer durchführen? 789
Wie kann ich meinen Amazon ECS-Service während einer Bereitstellung mit neuen Informationen aktualisieren? 789
Fehlerbehebung AWS Probleme bei der Lambda-Bereitstellung 790
AWS Lambda Bereitstellungen schlagen fehl, nachdem eine Lambda-Bereitstellung, für die keine Rollbacks konfiguriert wurden, manuell gestoppt wurde 790
Beheben von Problemen bei Bereitstellungsgruppen 791
Durch Markieren einer Instance als Teil einer Bereitstellungsgruppe wird Ihre Anwendung nicht automatisch in der neuen Instance bereitgestellt 791
Beheben von Instance-Problemen 791
Tags müssen korrekt festgelegt sein 792
AWS CodeDeploy Der Agent muss auf Instanzen installiert sein und ausgeführt werden 792
Bereitstellungen schlagen für bis zu einer Stunde nicht fehl, nachdem eine Instance während einer Bereitstellung beendet wurde 792
Analysieren von Protokolldateien zur Untersuchung von Bereitstellungsfehlern auf Instances 793
Erstellen Sie eine neue CodeDeploy Protokolldatei, falls sie versehentlich gelöscht wurde .. 793
Problembehandlung bei Bereitstellungsfehlern „InvalidSignatureException — Signatur ist abgelaufen: [Zeit] ist jetzt vor [Zeit]“ 794
Beheben Sie GitHub Token-Probleme 794
Ungültiges GitHub OAuth-Token 794
GitHub Die maximale Anzahl von OAuth-Token wurde überschritten 795
Probleme mit Amazon EC2 Auto Scaling beheben 795
Allgemeine Amazon EC2 Auto Scaling Scaling-Fehlerbehebung 796

"gibt Ihnen CodeDeployRole nicht die Erlaubnis, folgende Operationen auszuführen AWS Service: AmazonAutoScaling "Fehler ... 797
Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe werden kontinuierlich bereitgestellt und beendet, bevor eine Revision bereitgestellt werden kann ... 798
Das Beenden oder Neustarten einer Amazon EC2 Auto Scaling Scaling-Instance kann dazu führen, dass Bereitstellungen fehlschlagen ... 799
Vermeiden Sie es, mehrere Bereitstellungsgruppen einer einzigen Amazon EC2 Auto Scaling-Gruppe zuzuordnen ... 800
EC2-Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe können nicht gestartet werden und erhalten den Fehler „Heartbeat Timeout“ ... 800
Nicht übereinstimmende Amazon EC2 Auto Scaling Scaling-Lifecycle-Hooks können dazu führen, dass automatische Bereitstellungen für Amazon EC2 Auto Scaling Scaling-Gruppen gestoppt werden oder fehlschlagen ... 803
Fehler „Die Bereitstellung ist fehlgeschlagen, weil keine Instances für Ihre Bereitstellungsgruppe gefunden wurden“ ... 805
Fehlercodes ... 813
Verwandte Themen ... 819
Ressourcen ... 820
Referenzhandbücher und Support-Ressourcen ... 820
Beispiele ... 820
Blogs ... 820
AWS Kits und Tools für die Softwareentwicklung ... 820
Dokumentverlauf ... 822
Frühere Aktualisierungen ... 843
AWS Glossar ... 869
... dccclxx

Was ist CodeDeploy?

CodeDeploy ist ein Bereitstellungsservice, der Anwendungsbereitstellungen auf Amazon EC2 EC2-Instances, lokalen Instances, serverlosen Lambda-Funktionen oder Amazon ECS-Services automatisiert.

Sie können unterschiedlichste Anwendungsinhalte bereitstellen, darunter:

- Code
- Serverlose Funktionen AWS Lambda
- Web- und Konfigurationsdateien
- Ausführbare Dateien
- Pakete
- Skripts
- Multimediadateien

CodeDeploy kann Anwendungsinhalte bereitstellen, die auf einem Server ausgeführt und in Amazon S3 S3-Buckets, GitHub Repositorys oder Bitbucket-Repositorys gespeichert sind. CodeDeploy kann auch eine serverlose Lambda-Funktion bereitstellen. Sie müssen den vorhandenen Code nicht verändern, um CodeDeploy verwenden zu können.

CodeDeploy macht es einfacher für Sie:

- Schnell neue Funktionen zu veröffentlichen.
- Aktualisieren Sie AWS Lambda Funktionsversionen.
- Ausfallzeiten während der Anwendungsbereitstellung zu vermeiden.
- Die Komplexität der Aktualisierung Ihrer Anwendungen zu beherrschen, ohne viele der Risiken, die mit fehleranfälligen manuellen Bereitstellungen verbunden sind.

Der Service passt sich der Größe Ihrer Infrastruktur an, sodass Sie mühelos eine einzige Instance oder Tausende von Instances bereitstellen können.

CodeDeploy funktioniert mit verschiedenen Systemen für Konfigurationsmanagement, Quellcodeverwaltung, kontinuierliche Integration, kontinuierliche Bereitstellung und kontinuierliche Bereitstellung. Weitere Informationen finden Sie unter Produktintegrationen.

Die CodeDeploy Konsole bietet auch eine Möglichkeit, schnell nach Ihren Ressourcen wie Repositorys, Build-Projekten, Bereitstellungsanwendungen und Pipelines zu suchen. Wählen Sie Go to Ressource (Zur Ressource) oder drücken Sie die Taste / und geben Sie dann den Namen der Ressource ein. Alle Übereinstimmungen werden in der Liste angezeigt. Bei der Suche wird nicht zwischen Groß- und Kleinschreibung unterschieden. Sie sehen nur die Ressourcen, für die Sie die Berechtigung besitzen. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für AWS CodeDeploy.

Themen

- Vorteile von AWS CodeDeploy
- Überblick über CodeDeploy Rechenplattformen
- Überblick über die CodeDeploy Bereitstellungstypen
- Bitte geben Sie uns Feedback
- CodeDeploy Hauptbestandteile
- CodeDeploy Bereitstellungen
- CodeDeploy Anwendungsspezifikationsdateien (AppSpec)

Vorteile von AWS CodeDeploy

CodeDeploy bietet diese Vorteile:

- Server-, serverlose und Container-Anwendungen. CodeDeployermöglicht es Ihnen, sowohl herkömmliche Anwendungen auf Servern als auch Anwendungen bereitzustellen, die eine serverlose AWS Lambda Funktionsversion oder eine Amazon ECS-Anwendung bereitstellen.
- Automatisierte Bereitstellungen. CodeDeploy automatisiert Ihre Anwendungsbereitstellungen in Ihren Entwicklungs-, Test- und Produktionsumgebungen vollständig. CodeDeploypasst sich Ihrer Infrastruktur an, sodass Sie die Bereitstellung auf einer oder auf Tausenden von Instanzen durchführen können.
- Minimierung von Ausfallzeit. Wenn Ihre Anwendung die EC2/On-Premises Rechenplattform verwendet, CodeDeploy trägt dies dazu bei, die Verfügbarkeit Ihrer Anwendung zu maximieren. CodeDeploy Führt während einer In-Place-Bereitstellung ein fortlaufendes Update für alle Amazon EC2 EC2-Instances durch. Sie können die Anzahl der Instances angeben, die für Updates gleichzeitig offline geschaltet werden. Während einer blue/green Bereitstellung wird die neueste Anwendungsversion auf Ersatz-Instances installiert. Der Datenverkehr wird wahlweise entweder sofort oder nach Beendigung der Tests der neuen Umgebung an diese Instances umgeleitet. Bei

beiden Bereitstellungstypen prüft und verfolgt CodeDeploy die Anwendungsintegrität nach den von Ihnen konfigurierten Regeln.

- Stoppen und Rollback. Sie können Bereitstellungen automatisch oder manuell stoppen und ein Rollback ausführen, wenn Fehler auftreten.
- Zentrale Steuerung. Sie können Ihre Bereitstellungen über die CodeDeploy Konsole oder die starten und deren Status verfolgen. AWS CLI Sie erhalten einen Bericht, in dem aufgeführt ist, wann jede Anwendungsrevision bereitgestellt wurde und für welche Amazon EC2 EC2-Instances.
- Einfach zu übernehmen. CodeDeploy ist plattformunabhängig und funktioniert mit jeder Anwendung. Sie können Ihren Setup-Code problemlos wiederverwenden. CodeDeploy kann auch in Ihren Software-Release-Prozess oder Ihre Continuous-Delivery-Toolchain integriert werden.
- Gleichzeitige Bereitstellungen. Wenn Sie mehr als eine Anwendung haben, die die EC2/On-Premises Rechenplattform verwendet, CodeDeploy können Sie sie gleichzeitig auf derselben Gruppe von Instanzen bereitstellen.

Überblick über CodeDeploy Rechenplattformen

CodeDeploy ist in der Lage, Anwendungen auf drei Computerplattformen bereitzustellen:

- EC2/On-Premises: Beschreibt Instanzen von physischen Servern, bei denen es sich um Amazon EC2 EC2-Cloud-Instances, lokale Server oder beides handeln kann. Anwendungen, die mit der EC2/On-Premises Rechenplattform erstellt wurden, können aus ausführbaren Dateien, Konfigurationsdateien, Bildern und mehr bestehen.

  Bereitstellungen, die die EC2/On-Premises Rechenplattform verwenden, verwalten die Art und Weise, wie der Datenverkehr zu den Instances geleitet wird, indem sie einen In-Place- oder blue/green Bereitstellungstyp verwenden. Weitere Informationen finden Sie unter Überblick über die CodeDeploy Bereitstellungstypen.
- AWS Lambda: Wird verwendet, um Anwendungen bereitzustellen, die aus einer aktualisierten Version einer Lambda-Funktion bestehen. AWS Lambda verwaltet die Lambda-Funktion in einer serverlosen Rechenumgebung, die aus einer Rechenstruktur mit hoher Verfügbarkeit besteht. Die gesamte Verwaltung der Rechenressourcen erfolgt durch. AWS Lambda Weitere Informationen finden Sie unter Serverloses Computing und Anwendungen. Weitere Hinweise zu AWS Lambda und Lambda-Funktionen finden Sie unter AWS Lambda.

Sie können die Art und Weise verwalten, wie der Datenverkehr während einer Bereitstellung auf die aktualisierten Lambda-Funktionsversionen umgeleitet wird, indem Sie eine kanarische, lineare oder All-in-Once-Konfiguration wählen.

- Amazon ECS: Wird verwendet, um eine containerisierte Amazon ECS-Anwendung als Task-Set bereitzustellen. CodeDeploy führt eine blue/green Bereitstellung durch, indem eine aktualisierte Version der Anwendung als neuer Ersatz-Aufgabensatz installiert wird. CodeDeploy leitet den Produktionsdatenverkehr vom ursprünglichen Anwendungs-Tasksatz zum Ersatz-Tasksatz um. Der ursprüngliche Tasksatz wird nach einer erfolgreichen Bereitstellung beendet. Weitere Informationen zu Amazon ECS finden Sie unter Amazon Elastic Container Service.

  Sie können die Art und Weise, wie der Datenverkehr während einer Bereitstellung auf den aktualisierten Aufgabensatz verschoben wird, verwalten, indem Sie eine Canary-, lineare oder All-at-once Konfiguration auswählen.

  > **Note**
  >
  > Amazon blue/green ECS-Bereitstellungen werden sowohl mit als auch CodeDeploy unterstützt. CloudFormation Details zu diesen Bereitstellungen werden in den nachfolgenden Abschnitten beschrieben.

In der folgenden Tabelle wird beschrieben, wie CodeDeploy Komponenten auf den einzelnen Rechenplattformen verwendet werden. Weitere Informationen finden Sie unter:

- Arbeiten mit Bereitstellungsgruppen in CodeDeploy
- Arbeiten mit Bereitstellungen in CodeDeploy
- Arbeiten mit Bereitstellungskonfigurationen in CodeDeploy
- Arbeiten mit Anwendungsrevisionen für CodeDeploy
- Arbeiten mit Anwendungen in CodeDeploy

CodeDeploy Komponente	EC2/On-Premises	AWS Lambda	Amazon ECS
Bereitstellungsgruppe	Stellt eine Revision für eine Gruppe von Instances bereit.	Stellt eine neue Version einer serverlosen Lambda-Funktion auf einer Recheninfrastruktur mit hoher Verfügbarkeit bereit.	Spezifiziert den Amazon ECS-Service mit der containerisierten Anwendung, die als Task-Set bereitgestellt werden soll, einem Produktions- und optionalen Test-Listener, der verwendet wird, um den Datenverkehr an die bereitgestellte Anwendung weiterzuleiten und zu beenden, wann der ursprüngliche

CodeDeploy Komponente	EC2/On-Premises	AWS Lambda	Amazon ECS
			Tasksatz der bereitgestellten Anwendung umgeleitet und beendet werden soll, sowie optionale Trigger-, Alarm- und Rollback-Einstellungen.

CodeDeploy Komponente	EC2/On-Premises	AWS Lambda	Amazon ECS
Bereitstellung	Stellt eine neue Version bereit, die aus einer Anwendung und einer Datei besteht. AppSpec Das AppSpec gibt an, wie die Anwendung auf den Instanzen in einer Bereitstellungsgruppe bereitgestellt wird.	Verschiebt den Produktionsdatenverkehr von einer Version einer Lambda-Funktion auf eine neue Version derselben Funktion. Die AppSpec Datei gibt an, welche Lambda-Funktionsversion bereitgestellt werden soll.	Stellt eine aktualisierte Version einer containerisierten Amazon ECS-Anwendung als neuen Ersatz-Tasksatz bereit. CodeDeploy leitet den Produktionsdatenverkehr vom Task-Set mit der Originalversion zum neuen Ersatz-Task-Set mit der aktualisierten Version um. Wenn die

CodeDeploy Komponente	EC2/On-Premises	AWS Lambda	Amazon ECS
			Bereitstellung abgeschlossen ist, wird der ursprüngliche Aufgabensatz beendet.
Bereitstellungskonfiguration	Einstellungen, die die Bereitstellungsgeschwindigkeit angeben, ebenso wie die Mindestanzahl der fehlerfreien Instances zu jedem Zeitpunkt während einer Bereitstellung.	Einstellungen, die bestimmen, wie der Verkehr auf die aktualisierten Lambda-Funktionsversionen verlagert wird.	Einstellungen, die bestimmen, wie der Datenverkehr in den aktualisierten Amazon ECS-Aufgabensatz verlagert wird.

CodeDeploy Komponente	EC2/On-Premises	AWS Lambda	Amazon ECS
Revision	Eine Kombination aus einer AppSpec Datei und Anwendungsdateien, z. B. ausführbaren Dateien, Konfigurationsdateien usw.	Eine AppSpec Datei, die angibt, welche Lambda-Funktion bereitgestellt werden soll, und Lambda-Funktionen, die während Event-Hooks im Deployment-Lebenszyklus Validierungstests ausführen können.	Eine AppSpec Datei, die Folgendes spezifiziert: • Die Amazon ECS-Aufgabendefinition für den Amazon ECS-Service mit der bereitzustellenden containerisierten Anwendung. • Den Container, in dem Ihre aktualisierte Anwendung bereitgestellt wird.

CodeDeploy Komponente	EC2/On-Premises	AWS Lambda	Amazon ECS
			• Einen Port für den Container, wohin der Produktionsdatenverkehr umgeleitet wird. • Optionale Netzwerkkonfigurationseinstellungen und Lambda-Funktionen, mit denen Validierungstests während Event-Hooks im Bereitstellungslebenszyklus ausgeführ

CodeDeploy Komponente	EC2/On-Premises	AWS Lambda	Amazon ECS
			t werden können.
Anwendung	Eine Sammlung von Bereitstellungsgruppen und Revisionen. Eine EC2/On-Premises Anwendung verwendet die EC2/On-Premises Rechenplattform.	Eine Sammlung von Bereitstellungsgruppen und Revisionen. Eine für eine AWS Lambda Bereitstellung verwendete Anwendung verwendet die serverlose AWS Lambda-Rechenplattform.	Eine Sammlung von Bereitstellungsgruppen und Revisionen. Eine Anwendung, die für eine Amazon ECS-Bereitstellung verwendet wird, verwendet die Amazon ECS-Rechenplattform.

Überblick über die CodeDeploy Bereitstellungstypen

CodeDeploy bietet zwei Optionen für Bereitstellungstypen:

- In-place Bereitstellung: Die Anwendung auf jeder Instanz in der Bereitstellungsgruppe wird gestoppt, die neueste Anwendungsversion wird installiert und die neue Version der Anwendung wird gestartet und validiert. Sie können einen Load Balancer verwenden, sodass jede Instanz während der Bereitstellung deregistriert und nach Abschluss der Bereitstellung wieder

betriebsbereit ist. Nur Bereitstellungen, die die EC2/On-Premises Rechenplattform verwenden, können direkte Bereitstellungen verwenden. Weitere Informationen zu In-Situ-Bereitstellungen finden Sie unter Überblick über eine direkte Bereitstellung.

> **Note**
>
> AWS Lambda- und Amazon ECS-Bereitstellungen können keinen In-Place-Bereitstellungstyp verwenden.

- Blue/green Bereitstellung: Das Verhalten Ihrer Bereitstellung hängt davon ab, welche Rechenplattform Sie verwenden:
  - Blue/green auf einer EC2/On-Premises Rechenplattform: Die Instanzen in einer Bereitstellungsgruppe (der ursprünglichen Umgebung) werden mithilfe der folgenden Schritte durch eine andere Gruppe von Instanzen (die Ersatzumgebung) ersetzt:
    - Instanzen werden für die Ersatzumgebung bereitgestellt.
    - Die neueste Anwendungsversion ist auf den Ersatzinstanzen installiert.
    - Für Aktivitäten wie Anwendungstests und Systemverifizierung fällt eine optionale Wartezeit an.
    - Instances in der Ersatzumgebung werden bei einem oder mehreren Elastic Load Balancing Balancing-Load Balancing-Load Balancern registriert, wodurch der Datenverkehr zu ihnen umgeleitet wird. Instances in der ursprünglichen Umgebung werden deregistriert und können beendet oder für andere Zwecke weiterlaufen.

    > **Note**
    >
    > Wenn Sie eine EC2/On-Premises Rechenplattform verwenden, beachten Sie, dass blue/green Bereitstellungen nur mit Amazon EC2 EC2-Instances funktionieren.

  - Blue/green auf einer AWS Lambda oder Amazon ECS-Rechenplattform: Der Datenverkehr wird schrittweise entsprechend einer kanarischen, linearen oder All-in-Once-Bereitstellungskonfiguration verschoben.
  - Blue/green Bereitstellungen durch CloudFormation: Der Datenverkehr wird im Rahmen eines Stack-Updates von Ihren aktuellen Ressourcen auf Ihre aktualisierten Ressourcen verlagert. CloudFormation Derzeit werden nur blue/green ECS-Bereitstellungen unterstützt.

Weitere Informationen zu blue/green Bereitstellungen finden Sie unter. Überblick über eine Bereitstellung blue/green

> **Note**
>
> Mit dem CodeDeploy Agenten können Sie eine Bereitstellung auf einer Instance durchführen, bei der Sie angemeldet sind, ohne dass Sie eine Anwendung, eine Bereitstellungsgruppe oder sogar ein AWS Konto benötigen. Weitere Informationen finden Sie unter Verwenden Sie den CodeDeploy Agenten, um ein Bereitstellungspaket auf einem lokalen Computer zu validieren.

Themen

- Überblick über eine direkte Bereitstellung
- Überblick über eine Bereitstellung blue/green

Überblick über eine direkte Bereitstellung

> **Note**
>
> AWS Lambda- und Amazon ECS-Bereitstellungen können keinen In-Place-Bereitstellungstyp verwenden.

So funktioniert eine direkte Bereitstellung:

1. Zunächst erstellen Sie bereitstellbaren Inhalt auf Ihrem lokalen Entwicklungscomputer oder einer ähnlichen Umgebung und fügen dann eine Anwendungsspezifikationsdatei (AppSpec Datei) hinzu. Die AppSpec Datei ist einzigartig für CodeDeploy. Sie definiert die Bereitstellungsaktionen, die Sie ausführen CodeDeploy möchten. Sie bündeln Ihre bereitstellbaren Inhalte und die AppSpec Datei in einer Archivdatei und laden sie dann in einen Amazon S3 S3-Bucket oder ein GitHub Repository hoch. Diese Archivdatei wird als Anwendungsrevision (oder einfach Revision) bezeichnet.
2. Als Nächstes geben Sie Informationen CodeDeploy zu Ihrer Bereitstellung an, z. B. aus welchem Amazon S3 S3-Bucket oder GitHub -Repository die Revision abgerufen werden soll und für welche Gruppe von Amazon EC2 EC2-Instances der Inhalt bereitgestellt werden soll. CodeDeploy nennt eine Reihe von Amazon EC2 EC2-Instances eine Bereitstellungsgruppe. Eine Bereitstellungsgruppe enthält individuell markierte Amazon EC2-Instances, Amazon EC2 EC2-Instances in Amazon EC2 Auto Scaling-Gruppen oder beides.

Jedes Mal, wenn Sie eine neue Anwendungsrevision, die für die Bereitstellungsgruppe bereitgestellt werden soll, erfolgreich hochladen, wird dieses Paket als Zielrevision für die Bereitstellungsgruppe festgelegt. Mit anderen Worten, die Anwendungsrevision, die aktuell zur Bereitstellung vorgesehen ist, ist die Zielrevision. Diese Revision wird auch für automatische Bereitstellungen verwendet.

3. Als Nächstes fragt der CodeDeploy Agent auf jeder Instance ab CodeDeploy , was und wann aus dem angegebenen Amazon S3 S3-Bucket oder GitHub Repository abgerufen werden soll.
4. Schließlich ruft der CodeDeploy Agent auf jeder Instance die Zielversion aus dem Amazon S3 S3-Bucket oder GitHub Repository ab und stellt den Inhalt anhand der Anweisungen in der AppSpec Datei für die Instance bereit.

CodeDeploy zeichnet Ihre Bereitstellungen auf, sodass Sie den Bereitstellungsstatus, die Bereitstellungskonfigurationsparameter, den Zustand der Instance usw. abrufen können.

Überblick über eine Bereitstellung blue/green

Eine blue/green Bereitstellung dient dazu, Ihre Anwendungen zu aktualisieren und gleichzeitig Unterbrechungen zu minimieren, die durch die Änderungen einer neuen Anwendungsversion verursacht werden. CodeDeploy stellt Ihre neue Anwendungsversion zusammen mit der alten Version bereit, bevor Ihr Produktionsdatenverkehr umgeleitet wird.

- AWS Lambda: Der Verkehr wird von einer Version einer Lambda-Funktion auf eine neue Version derselben Lambda-Funktion umgestellt.
- Amazon ECS: Der Datenverkehr wird von einem Task-Set in Ihrem Amazon ECS-Service auf ein aktualisiertes Ersatz-Task-Set im selben Amazon ECS-Service umgeleitet.
- EC2/On-Premises: Der Datenverkehr wird von einer Gruppe von Instances in der ursprünglichen Umgebung auf einen neuen Satz von Instances verlagert.

Alle AWS Lambda- und Amazon ECS-Bereitstellungen sind. blue/green Eine EC2/On-Premises Bereitstellung kann vor Ort oder vor Ort erfolgen. blue/green Eine blue/green Bereitstellung bietet eine Reihe von Vorteilen gegenüber einer Bereitstellung vor Ort:

- Sie können eine Anwendung in der neuen Ersatzumgebung installieren und testen und für die Produktion bereitstellen, indem Sie den Datenverkehr einfach umleiten.

- Wenn Sie die EC2/On-Premises Rechenplattform verwenden, ist die Rückkehr zur neuesten Version einer Anwendung schneller und zuverlässiger. Dies liegt daran, dass der Verkehr wieder an die ursprünglichen Instances umgeleitet werden kann, solange diese noch nicht beendet wurden. Bei einer In-Situ-Bereitstellung muss dagegen ein Rollback der Versionen durchgeführt werden, indem die vorherige Anwendungsversion bereitgestellt wird.
- Wenn Sie die EC2/On-Premises Rechenplattform verwenden, werden neue Instanzen für eine blue/green Bereitstellung bereitgestellt und entsprechen den aktuellsten Serverkonfigurationen. Auf diese Weise können Sie Probleme vermeiden, die manchmal bei lang laufenden Instances auftreten.
- Wenn Sie die AWS Lambda-Rechenplattform verwenden, kontrollieren Sie, wie der Datenverkehr von Ihrer ursprünglichen AWS Lambda-Funktionsversion auf Ihre neue AWS Lambda-Funktionsversion übertragen wird.
- Wenn Sie die Amazon ECS-Rechenplattform verwenden, kontrollieren Sie, wie der Datenverkehr von Ihrem ursprünglichen Aufgabensatz auf Ihren neuen Aufgabensatz verlagert wird.

Bei einer blue/green Bereitstellung mit CloudFormation kann eine der folgenden Methoden verwendet werden:

- CloudFormation Vorlagen für Bereitstellungen: Wenn Sie Bereitstellungen mit CloudFormation Vorlagen konfigurieren, werden Ihre Bereitstellungen durch Updates ausgelöst. CloudFormation Wenn Sie eine Ressource ändern und eine Vorlagenänderung hochladen, wird durch ein Stack-Update die neue Bereitstellung CloudFormation initiiert. Eine Liste der Ressourcen, die Sie in CloudFormation Vorlagen verwenden können, finden Sie unterCloudFormation Vorlagen als CodeDeploy Referenz.
- Blue/green Bereitstellungen durch CloudFormation: Sie können es verwenden CloudFormation , um Ihre blue/green Bereitstellungen mithilfe von Stack-Updates zu verwalten. Sie definieren sowohl die blauen als auch die grünen Ressourcen zusätzlich zu den Einstellungen für das Routing und die Stabilisierung des Datenverkehrs innerhalb der Stack-Vorlage. Wenn Sie anschließend ausgewählte Ressourcen während eines Stack-Updates aktualisieren, werden alle erforderlichen grünen Ressourcen CloudFormation generiert, der Datenverkehr auf der Grundlage der angegebenen Datenverkehrs-Routing-Parameter verschoben und die blauen Ressourcen gelöscht. Weitere Informationen finden Sie unter Automatisieren von Amazon blue/green ECS-Bereitstellungen CodeDeploy mithilfe CloudFormation im AWS CloudFormation Benutzerhandbuch.

> **Note**
>
> Wird nur für Amazon blue/green ECS-Bereitstellungen unterstützt.

Wie Sie eine blue/green Bereitstellung konfigurieren, hängt davon ab, welche Rechenplattform Ihre Bereitstellung verwendet.

Blue/Green Bereitstellung auf einem AWS Lambda oder Amazon ECS-Rechenplattform

Wenn Sie die AWS Lambda oder die Amazon ECS-Rechenplattform verwenden, müssen Sie angeben, wie der Datenverkehr von der ursprünglichen AWS Lambda Funktion oder dem Amazon ECS-Aufgabensatz auf den neuen Funktions- oder Tasksatz verlagert wird. Um anzugeben, wie der Datenverkehr verlagert wird, müssen Sie eine der folgenden Bereitstellungskonfigurationen angeben:

- kanarisch
- linear
- alles auf einmal

Informationen darüber, wie der Datenverkehr in einer kanarischen, linearen oder All-at-Once-Bereitstellungskonfiguration verlagert wird, finden Sie unter. Bereitstellungskonfiguration

Einzelheiten zur Konfiguration der Lambda-Bereitstellung finden Sie unterBereitstellungskonfigurationen auf einem AWS Lambda Plattform für die Datenverarbeitung.

Einzelheiten zur Amazon ECS-Bereitstellungskonfiguration finden Sie unterBereitstellungskonfigurationen auf einer Amazon-ECS-Datenverarbeitungsplattform.

Blue/Green Bereitstellung auf einer EC2/on-premises Rechenplattform

> **Note**
>
> Sie müssen Amazon EC2 EC2-Instances für blue/green Bereitstellungen auf der EC2/On-Premises Rechenplattform verwenden. On-premises Instances werden für den blue/green Bereitstellungstyp nicht unterstützt.

Wenn Sie die EC2/On-Premises Rechenplattform verwenden, gilt Folgendes:

Sie benötigen eine oder mehrere Amazon EC2-Instances mit identifizierenden Amazon EC2-Tags oder eine Amazon EC2 Auto Scaling-Gruppe. Die Instances müssen zusätzlich die folgenden Anforderungen erfüllen:

- An jede Amazon EC2 EC2-Instance muss das richtige IAM-Instance-Profil angehängt sein.
- Der CodeDeploy Agent muss auf jeder Instance installiert sein und ausgeführt werden.

> **Note**
>
> In der Regel wird auf den Instanzen in Ihrer ursprünglichen Umgebung auch eine Anwendungsversion ausgeführt, dies ist jedoch keine Voraussetzung für eine blue/green Bereitstellung.

Wenn Sie eine Bereitstellungsgruppe erstellen, die in blue/green Bereitstellungen verwendet wird, können Sie wählen, wie Ihre Ersatzumgebung angegeben wird:

Kopieren Sie eine bestehende Amazon EC2 Auto Scaling Scaling-Gruppe: CodeDeploy Erstellt während der blue/green Bereitstellung die Instances für Ihre Ersatzumgebung während der Bereitstellung. Bei dieser Option wird die von Ihnen angegebene Amazon EC2 Auto Scaling Scaling-Gruppe als Vorlage für die Ersatzumgebung CodeDeploy verwendet, einschließlich derselben Anzahl laufender Instances und vieler anderer Konfigurationsoptionen.

Wählen Sie Instances manuell aus: Sie können die Instances, die als Ihr Ersatz gezählt werden sollen, mithilfe von Amazon EC2-Instance-Tags, Amazon EC2 Auto Scaling-Gruppennamen oder beidem angeben. Wenn Sie diese Option wählen, müssen Sie die Instances für die Ersatzumgebung erst beim Erstellen der Bereitstellung angeben.

Funktionsweise:

1. Sie haben bereits Instances oder eine Amazon EC2 Auto Scaling Scaling-Gruppe, die als Ihre ursprüngliche Umgebung dient. Wenn Sie eine blue/green Bereitstellung zum ersten Mal ausführen, verwenden Sie in der Regel Instances, die bereits in einer In-Place-Bereitstellung verwendet wurden.

2. In einer vorhandenen CodeDeploy Anwendung erstellen Sie eine blue/green Bereitstellungsgruppe, in der Sie zusätzlich zu den Optionen, die für eine direkte Bereitstellung erforderlich sind, Folgendes angeben:
   - Der oder die Load Balancer, die während des Bereitstellungsprozesses den Datenverkehr von Ihrer ursprünglichen Umgebung an Ihre Ersatzumgebung weiterleiten. blue/green
   - Ob der Datenverkehr sofort an die Ersatzumgebung umgeleitet oder gewartet werden soll, bis Sie den Datenverkehr manuell umleiten.
   - Die Übertragungsgeschwindigkeit, mit welcher der Datenverkehr an die Ersatz-Instances umgeleitet werden soll.
   - Ob die Instances, die ersetzt werden, beendet oder weiterhin ausgeführt werden sollen.
3. Sie erstellen eine Bereitstellung für diese Bereitstellungsgruppe, in deren Verlauf Folgendes geschieht:
   a. Wenn Sie sich dafür entscheiden, eine Amazon EC2 Auto Scaling Scaling-Gruppe zu kopieren, werden Instances für Ihre Ersatzumgebung bereitgestellt.
   b. Die Anwendungsrevision, die Sie für die Bereitstellung angeben, wird auf den Ersatz-Instances installiert.
   c. Wenn Sie in den Bereitstellungsgruppeneinstellungen eine Wartezeit angeben, wird die Bereitstellung zeitweilig unterbrochen. In dieser Zeit können Sie die Ersatzumgebung testen und überprüfen. Wenn Sie den Datenverkehr nicht vor Ablauf der Wartezeit manuell umleiten, wird die Bereitstellung gestoppt.
   d. Instances in der Ersatzumgebung werden bei einem Elastic Load Balancing Load Balancer registriert und der Datenverkehr wird an sie weitergeleitet.
   e. Die Registrierung der ursprünglichen Umgebung wird aufgehoben, und diese Instances werden entsprechend der Angaben in der Bereitstellungsgruppe entweder beendet oder weiterhin ausgeführt.

Blue/Green Bereitstellung durch CloudFormation

Sie können CodeDeploy blue/green Bereitstellungen verwalten, indem Sie Ihre Ressourcen mit einer CloudFormation Vorlage modellieren.

Wenn Sie Ihre blue/green Ressourcen mithilfe einer CloudFormation Vorlage modellieren, erstellen Sie ein Stack-Update CloudFormation , das Ihren Aufgabensatz aktualisiert. Der Produktionsdatenverkehr verschiebt sich vom ursprünglichen Aufgabensatz Ihres Services in einen Ersatzaufgabensatz entweder auf einmal, mit linearen Bereitstellungen und Bake-Zeit, oder mit

Canary-Bereitstellungen. Die Stack-Aktualisierung initiiert eine Bereitstellung in CodeDeploy. Sie können den Status und den Verlauf der Bereitstellung in einsehen CodeDeploy, aber Sie erstellen oder verwalten keine CodeDeploy Ressourcen außerhalb der CloudFormation Vorlage.

> **Note**
>
> Für blue/green Bereitstellungen bis CloudFormation erstellen Sie keine CodeDeploy Anwendung oder Bereitstellungsgruppe.

Diese Methode unterstützt nur Amazon blue/green ECS-Bereitstellungen. Weitere Informationen zu blue/green Bereitstellungen bis finden Sie CloudFormation unter. Erstellen Sie eine Amazon blue/green ECS-Bereitstellung über CloudFormation

Bitte geben Sie uns Feedback

Wir freuen uns über Ihr Feedback. Um uns zu kontaktieren, besuchen Sie das CodeDeploy Forum.

Topics

- Primary Components
- Deployments
- Application Specification Files

CodeDeploy Hauptbestandteile

Bevor Sie mit der Arbeit mit dem Service beginnen, sollten Sie sich mit den wichtigsten Komponenten des CodeDeploy Bereitstellungsprozesses vertraut machen.

Themen

- Anwendung
- Plattform für die Datenverarbeitung
- Bereitstellungskonfiguration
- Bereitstellungsgruppe
- Bereitstellungstyp
- IAM-Instanzprofil

- Revision
- Service – -Rolle
- Zielrevision
- Andere Komponenten

Anwendung

Eine Anwendung ist ein Name, der die Anwendung, die Sie bereitstellen möchten, eindeutig identifiziert. CodeDeploy verwendet diesen Namen, der als Container fungiert, um sicherzustellen, dass während einer Bereitstellung auf die richtige Kombination aus Version, Bereitstellungskonfiguration und Bereitstellungsgruppe verwiesen wird.

Plattform für die Datenverarbeitung

Eine Rechenplattform ist eine Plattform, auf der eine Anwendung CodeDeploy bereitgestellt wird. Es gibt drei Rechenplattformen:

- EC2/On-Premises: Beschreibt Instanzen von physischen Servern, bei denen es sich um Amazon EC2 EC2-Cloud-Instances, lokale Server oder beides handeln kann. Mit der EC2/On-Premises Rechenplattform erstellte Anwendungen können aus ausführbaren Dateien, Konfigurationsdateien, Bildern und mehr bestehen.

  Bereitstellungen, die die EC2/On-Premises Rechenplattform verwenden, verwalten die Art und Weise, wie der Datenverkehr zu den Instanzen geleitet wird, indem sie einen direkten oder einen blue/green Bereitstellungstyp verwenden. Weitere Informationen finden Sie unter Überblick über die CodeDeploy Bereitstellungstypen.
- AWS Lambda: Wird verwendet, um Anwendungen bereitzustellen, die aus einer aktualisierten Version einer Lambda-Funktion bestehen. AWS Lambda verwaltet die Lambda-Funktion in einer serverlosen Rechenumgebung, die aus einer Rechenstruktur mit hoher Verfügbarkeit besteht. Die gesamte Verwaltung der Rechenressourcen erfolgt durch. AWS Lambda Weitere Informationen finden Sie unter Serverloses Computing und Anwendungen. Weitere Hinweise zu AWS Lambda und Lambda-Funktionen finden Sie unter AWS Lambda.

  Sie können die Art und Weise verwalten, wie der Datenverkehr während einer Bereitstellung auf die aktualisierten Lambda-Funktionsversionen umgeleitet wird, indem Sie eine kanarische, lineare oder All-in-Once-Konfiguration wählen.

- Amazon ECS: Wird verwendet, um eine containerisierte Amazon ECS-Anwendung als Task-Set bereitzustellen. CodeDeploy führt eine blue/green Bereitstellung durch, indem eine aktualisierte Version der Anwendung als neuer Ersatz-Aufgabensatz installiert wird. CodeDeploy leitet den Produktionsdatenverkehr vom ursprünglichen Anwendungs-Tasksatz zum Ersatz-Tasksatz um. Der ursprüngliche Tasksatz wird nach einer erfolgreichen Bereitstellung beendet. Weitere Informationen zu Amazon ECS finden Sie unter Amazon Elastic Container Service.

  Sie können die Art und Weise, wie der Datenverkehr während einer Bereitstellung auf den aktualisierten Aufgabensatz verschoben wird, verwalten, indem Sie eine Canary-, lineare oder All-at-once Konfiguration auswählen.

> **Note**
>
> Amazon blue/green ECS-Bereitstellungen werden sowohl durch als auch CodeDeploy unterstützt. CloudFormation Details zu diesen Bereitstellungen werden in den nachfolgenden Abschnitten beschrieben.

Bereitstellungskonfiguration

Eine Bereitstellungskonfiguration besteht aus einer Reihe von Bereitstellungsregeln und Bedingungen für Erfolg und Misserfolg bei der Bereitstellung, die CodeDeploy während einer Bereitstellung verwendet werden. Wenn Ihre Bereitstellung die EC2/On-Premises Rechenplattform verwendet, können Sie die Mindestanzahl fehlerfreier Instanzen für die Bereitstellung angeben. Wenn Ihre Bereitstellung die AWS Lambda oder die Amazon ECS-Rechenplattform verwendet, können Sie angeben, wie der Datenverkehr an Ihre aktualisierte Lambda-Funktion oder Ihren aktualisierten ECS-Tasksatz weitergeleitet wird.

Weitere Informationen zur Angabe der Mindestanzahl fehlerfreier Hosts für eine Bereitstellung, die die EC2/On-Premises Rechenplattform verwendet, finden Sie unter. Ungefähr die Mindestanzahl fehlerfreier Instanzen

Die folgenden Bereitstellungskonfigurationen geben an, wie der Datenverkehr während einer Bereitstellung weitergeleitet wird, die die Lambda- oder die ECS-Rechenplattform verwendet:

- Canary: Der Datenverkehr wird in zwei Inkrementen verschoben. Sie können aus vordefinierten Canary-Optionen wählen, die den Prozentsatz des Datenverkehrs angeben, der im ersten Schritt

auf Ihre aktualisierte Lambda-Funktion oder Ihren ECS-Tasksatz übertragen wird, und das Intervall in Minuten, bevor der verbleibende Verkehr in der zweiten Stufe umgeleitet wird.

- Linear: Der Datenverkehr wird in gleich großen Inkrementen mit einer gleichen Anzahl von Minuten zwischen den Inkrementen verschoben. Sie können aus vordefinierten linearen Optionen wählen, die den prozentualen Anteil des Datenverkehrs angeben, der in jedem Inkrementschritt verschoben wird, sowie die Anzahl der Minuten zwischen den einzelnen Inkrementschritten.
- All-at-once: Der gesamte Datenverkehr wird auf einmal von der ursprünglichen Lambda-Funktion oder dem ursprünglichen ECS-Aufgabensatz auf den aktualisierten Funktions- oder Tasksatz umgestellt.

Bereitstellungsgruppe

Eine Bereitstellungsgruppe besteht aus einer Reihe von einzelnen Instanzen. Eine Bereitstellungsgruppe enthält individuell markierte Instances, Amazon EC2-Instances in Amazon EC2 Auto Scaling-Gruppen oder beides. Informationen zu Amazon EC2 EC2-Instance-Tags finden Sie unter Arbeiten mit Tags mithilfe der Konsole. Weitere Informationen über lokale Instances finden Sie unter Working with On-Premises Instances. Informationen zu Amazon EC2 Auto Scaling finden Sie unterIntegration CodeDeploy mit Amazon EC2 Auto Scaling.

Bereitstellungstyp

Ein Bereitstellungstyp ist eine Methode, mit der die neueste Anwendungsversion auf Instances in einer Bereitstellungsgruppe verfügbar gemacht wird. Es gibt zwei Bereitstellungstypen:

- In-place Bereitstellung: Die Anwendung auf jeder Instanz in der Bereitstellungsgruppe wird gestoppt, die neueste Anwendungsversion wird installiert und die neue Version der Anwendung wird gestartet und validiert. Sie können einen Load Balancer verwenden, sodass jede Instanz während der Bereitstellung deregistriert und nach Abschluss der Bereitstellung wieder betriebsbereit ist. Nur Bereitstellungen, die die EC2/On-Premises Rechenplattform verwenden, können direkte Bereitstellungen verwenden. Weitere Informationen zu In-Situ-Bereitstellungen finden Sie unter Überblick über eine direkte Bereitstellung.
- Blue/green Bereitstellung: Das Verhalten Ihrer Bereitstellung hängt davon ab, welche Rechenplattform Sie verwenden:
  - Blue/green auf einer EC2/On-Premises Rechenplattform: Die Instanzen in einer Bereitstellungsgruppe (der ursprünglichen Umgebung) werden mithilfe der folgenden Schritte durch eine andere Gruppe von Instanzen (die Ersatzumgebung) ersetzt:

- Instanzen werden für die Ersatzumgebung bereitgestellt.
- Die neueste Anwendungsversion ist auf den Ersatzinstanzen installiert.
- Für Aktivitäten wie Anwendungstests und Systemverifizierung fällt eine optionale Wartezeit an.
- Instances in der Ersatzumgebung werden bei einem oder mehreren Elastic Load Balancing Balancing-Load Balancing-Load Balancern registriert, wodurch der Datenverkehr zu ihnen umgeleitet wird. Instances in der ursprünglichen Umgebung werden deregistriert und können beendet oder für andere Zwecke weiterlaufen.

> **Note**
>
> Wenn Sie eine EC2/On-Premises Rechenplattform verwenden, beachten Sie, dass blue/green Bereitstellungen nur mit Amazon EC2 EC2-Instances funktionieren.

- Blue/green auf einer AWS Lambda oder Amazon ECS-Rechenplattform: Der Datenverkehr wird schrittweise entsprechend einer kanarischen, linearen oder All-in-Once-Bereitstellungskonfiguration verschoben.
- Blue/green Bereitstellungen durch CloudFormation: Der Datenverkehr wird im Rahmen eines Stack-Updates von Ihren aktuellen Ressourcen auf Ihre aktualisierten Ressourcen verlagert. CloudFormation Derzeit werden nur blue/green ECS-Bereitstellungen unterstützt.

Weitere Informationen zu blue/green Bereitstellungen finden Sie unter. Überblick über eine Bereitstellung blue/green

> **Note**
>
> Amazon blue/green ECS-Bereitstellungen werden sowohl mit als auch CodeDeploy unterstützt. CloudFormation Details zu diesen Bereitstellungen werden in den nachfolgenden Abschnitten beschrieben.

IAM-Instanzprofil

Ein IAM-Instance-Profil ist eine IAM-Rolle, die Sie Ihren Amazon EC2 EC2-Instances zuordnen. Dieses Profil beinhaltet die erforderlichen Berechtigungen für den Zugriff auf die Amazon S3 S3-Buckets oder GitHub Repositorys, in denen die Anwendungen gespeichert sind. Weitere

Informationen finden Sie unter Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances.

Revision

Eine Revision ist eine Version Ihrer Anwendung. Eine AWS Lambda-Bereitstellungsrevision ist eine YAML- oder JSON-formatted Datei, die Informationen über die bereitzustellende Lambda-Funktion spezifiziert. Eine EC2/On-Premises Bereitstellungsrevision ist eine Archivdatei, die Quellinhalte (Quellcode, Webseiten, ausführbare Dateien und Bereitstellungsskripts) und eine Anwendungsspezifikationsdatei (Datei) enthält. AppSpec AWS Lambda-Revisionen können in Amazon S3 S3-Buckets gespeichert werden. EC2/On-Premises Revisionen werden in Amazon S3 S3-Buckets oder GitHub Repositorys gespeichert. In Amazon S3 wird eine Revision eindeutig durch ihren Amazon-S3-Objektschlüssel und ihren ETag, die Version oder beides identifiziert. Denn GitHub eine Revision wird anhand ihrer Commit-ID eindeutig identifiziert.

Service – -Rolle

Eine Servicerolle ist eine IAM-Rolle, die einem AWS Dienst Berechtigungen erteilt, sodass dieser auf AWS Ressourcen zugreifen kann. Die Richtlinien, die Sie der Servicerolle zuordnen, bestimmen, auf welche AWS Ressourcen der Dienst zugreifen kann und welche Aktionen er mit diesen Ressourcen ausführen kann. Denn CodeDeploy eine Servicerolle wird für Folgendes verwendet:

- Um entweder die auf die Instances angewendeten Tags oder die Amazon EC2 Auto Scaling Scaling-Gruppennamen zu lesen, die den Instances zugeordnet sind. Auf diese Weise können CodeDeploy Instances identifiziert werden, auf denen Anwendungen bereitgestellt werden können.
- Um Operationen auf Instances, Amazon EC2 Auto Scaling Scaling-Gruppen und Elastic Load Balancing Balancing-Load Balancers durchzuführen.
- Um Informationen zu Amazon SNS SNS-Themen zu veröffentlichen, sodass Benachrichtigungen gesendet werden können, wenn bestimmte Bereitstellungs- oder Instance-Ereignisse eintreten.
- Zum Abrufen von Informationen über CloudWatch Alarme, um die Alarmüberwachung für Bereitstellungen einzurichten.

Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.

Zielrevision

Eine Zielrevision ist die neueste Version der Anwendungsrevision, die Sie in Ihr Repository hochgeladen haben und die Sie für die Instances in einer Bereitstellungsgruppe bereitstellen

möchten. Mit anderen Worten: Die Anwendungsrevision, die derzeit für die Bereitstellung vorgesehen ist. Diese Revision wird auch für automatische Bereitstellungen verwendet.

Andere Komponenten

Informationen zu anderen Komponenten des CodeDeploy Workflows finden Sie in den folgenden Themen:

- Wählen Sie einen CodeDeploy Repository-Typ
- Deployments
- Application Specification Files
- Instance Health
- Mit dem CodeDeploy Agenten arbeiten
- Working with On-Premises Instances

CodeDeploy Bereitstellungen

Dieses Thema bietet Informationen über die Komponenten und Workflows von Bereitstellungen in CodeDeploy. Der Bereitstellungsprozess variiert je nach Rechenplattform oder Bereitstellungsmethode (Lambda, Amazon ECS oder so AWS CloudFormation) EC2/On-Premises, die Sie für Ihre Bereitstellungen verwenden.

Themen

- Bereitstellungen auf einem AWS Lambda Compute-Plattform
- Bereitstellung auf einer Amazon-ECS-Datenverarbeitungsplattform
- Bereitstellungen auf einer Rechenplattform EC2/On-Premises

Bereitstellungen auf einem AWS Lambda Compute-Plattform

Dieses Thema enthält Informationen zu den Komponenten und zum Arbeitsablauf von CodeDeploy Bereitstellungen, die die AWS Lambda-Rechenplattform verwenden.

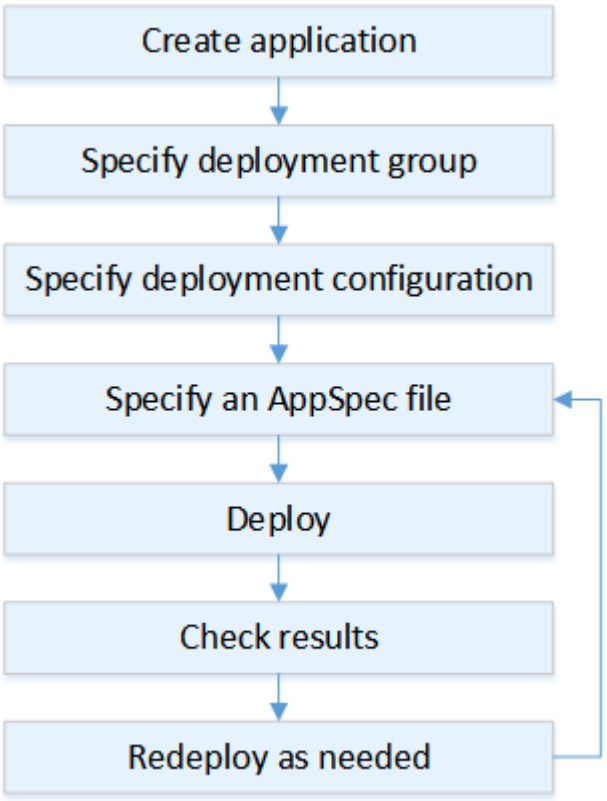
Themen

- Bereitstellungs-Workflow auf einem AWS Lambda Plattform für die Datenverarbeitung
- Die Revision Ihrer Anwendung wird hochgeladen

- Erstellen Sie Ihre Anwendungs- und Bereitstellungsgruppen
- Revision Ihrer Anwendung bereitstellen
- Aktualisierung Ihrer Anwendung
- Bereitstellungen wurden gestoppt und sind fehlgeschlagen
- Neubereitstellungen und Bereitstellungs-Rollbacks

Bereitstellungs-Workflow auf einem AWS Lambda Plattform für die Datenverarbeitung

Das folgende Diagramm zeigt die wichtigsten Schritte bei der Bereitstellung neuer und aktualisierter AWS Lambda -Funktionen.



Zu diesen Schritten gehören:

1. Erstellen Sie eine Anwendung und geben Sie ihr einen eindeutigen Namen für die Anwendungsrevisionen, die Sie bereitstellen möchten. Um Lambda-Funktionen bereitzustellen, wählen Sie bei der Erstellung Ihrer Anwendung die AWS Lambda-Rechenplattform aus. CodeDeploy verwendet diesen Namen während einer Bereitstellung, um sicherzustellen, dass

er auf die richtigen Bereitstellungskomponenten verweist, z. B. die Bereitstellungsgruppe, die Bereitstellungskonfiguration und die Anwendungsversion. Weitere Informationen finden Sie unter Erstellen Sie eine Anwendung mit CodeDeploy.

2. Richten Sie eine Bereitstellungsgruppe durch Angabe des Namens der Bereitstellungsgruppe ein.
3. Wählen Sie eine Bereitstellungskonfiguration, um anzugeben, wie der Datenverkehr von Ihrer ursprünglichen AWS Lambda Funktionsversion auf Ihre neue Lambda-Funktionsversion übertragen wird. Weitere Informationen finden Sie unter View Deployment Configuration Details.
4. Hochladen einer Anwendungsspezifikationsdatei (AppSpec Datei) auf Amazon S3. Die AppSpec Datei spezifiziert eine Lambda-Funktionsversion und Lambda-Funktionen, die zur Validierung Ihrer Bereitstellung verwendet werden. Wenn Sie keine AppSpec Datei erstellen möchten, können Sie mithilfe von YAML oder JSON direkt in der Konsole eine Lambda-Funktionsversion und Funktionen zur Validierung der Lambda-Bereitstellung angeben. Weitere Informationen finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.
5. Stellen Sie die Revision Ihrer Anwendung in der Bereitstellungsgruppe bereit. AWS CodeDeploy stellt die von Ihnen angegebene Lambda-Funktionsrevision bereit. Der Datenverkehr wird mithilfe der AppSpec Bereitstellungsdatei, die Sie bei der Erstellung Ihrer Anwendung ausgewählt haben, auf Ihre Lambda-Funktionsrevision umgeleitet. Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy.
6. Überprüfen Sie die Bereitstellungsergebnisse. Weitere Informationen finden Sie unter Überwachung von Bereitstellungen in CodeDeploy.

Die Revision Ihrer Anwendung wird hochgeladen

Platzieren Sie eine AppSpec Datei in Amazon S3 oder geben Sie sie direkt in die Konsole ein oder AWS CLI. Weitere Informationen finden Sie unter Application Specification Files.

Erstellen Sie Ihre Anwendungs- und Bereitstellungsgruppen

Eine CodeDeploy Bereitstellungsgruppe auf einer AWS Lambda-Rechenplattform identifiziert eine Sammlung von einer oder mehreren AppSpec Dateien. Jede AppSpec Datei kann eine Lambda-Funktionsversion bereitstellen. Eine Bereitstellungsgruppe definiert auch eine Reihe von Konfigurationsoptionen für zukünftige Bereitstellungen, wie z. B. Alarme und Rollback-Konfigurationen.

Revision Ihrer Anwendung bereitstellen

Jetzt sind Sie bereit, die in der AppSpec Datei angegebene Funktionsrevision für die Bereitstellungsgruppe bereitzustellen. Sie können die CodeDeploy Konsole oder den Befehl create-deployment verwenden. Es gibt Parameter, die Sie angeben können, um Ihre Bereitstellung zu steuern, einschließlich der Revision, Bereitstellungsgruppe und Bereitstellungskonfiguration.

Aktualisierung Ihrer Anwendung

Sie können Ihre Anwendung aktualisieren und dann die CodeDeploy Konsole verwenden oder den Befehl create-deployment aufrufen, um eine Version zu pushen.

Bereitstellungen wurden gestoppt und sind fehlgeschlagen

Sie können die CodeDeploy Konsole oder den Befehl stop-deployment verwenden, um eine Bereitstellung zu beenden. Wenn Sie versuchen, die Bereitstellung zu stoppen, gibt es drei Möglichkeiten:

- Die Bereitstellung wird beendet, und die Operation gibt den Status "Erfolgreich" zurück. In diesem Fall werden keine Bereitstellungslebenszyklusereignisse mehr für die Bereitstellungsgruppe für die gestoppte Bereitstellung ausgeführt.
- Die Bereitstellung wird nicht sofort beendet, und die Operation gibt den Status "Ausstehend" zurück. In diesem Fall werden einige Bereitstellungslebenszyklusereignisse in der Bereitstellungsgruppe weiterhin ausgeführt. Nachdem die ausstehende Operation abgeschlossen ist, geben nachfolgende Aufrufe zum Stoppen der Bereitstellung den Status "Erfolgreich" zurück.
- Die Bereitstellung kann nicht gestoppt werden, und der Vorgang gibt einen Fehler zurück. Weitere Informationen finden Sie unter ErrorInformationHäufige Fehler in der AWS CodeDeploy API-Referenz.

Genauso wie gestoppte Bereitstellungen können fehlgeschlagene Bereitstellungen dazu führen, dass einige Ereignisse im Bereitstellungslebenszyklus bereits ausgeführt wurden. Um herauszufinden, warum eine Bereitstellung fehlgeschlagen ist, verwenden Sie die CodeDeploy-Konsole oder analysieren Sie die Daten der Protokolldatei aus der fehlgeschlagenen Bereitstellung. Weitere Informationen erhalten Sie unter Revision der Anwendung und Bereinigung der Protokolldateien und Protokolldaten für CodeDeploy EC2/On-Premises Bereitstellungen anzeigen.

Neubereitstellungen und Bereitstellungs-Rollbacks

CodeDeploy implementiert Rollbacks, indem eine zuvor bereitgestellte Version als neue Bereitstellung erneut bereitgestellt wird.

Sie können eine Bereitstellungsgruppe für automatisches Rollback konfigurieren, wenn bestimmte Bedingungen erfüllt sind, einschließlich wenn eine Bereitstellung fehlschlägt oder ein Alarm-Überwachungsschwellenwert erreicht ist. Sie können auch die Rollback-Einstellungen für eine Bereitstellungsgruppe in einer einzelnen Bereitstellung überschreiben.

Sie können außerdem ein Rollback einer fehlgeschlagenen Bereitstellung durch erneute manuelle Bereitstellung einer zuvor bereitgestellten Revision durchführen.

In allen Fällen wird der neuen oder rückabgewickelten Bereitstellung eine eigene Bereitstellungs-ID zugewiesen. Die Liste der Bereitstellungen, die Sie in der CodeDeploy Konsole einsehen können, zeigt, welche das Ergebnis einer automatischen Bereitstellung sind.

Weitere Informationen finden Sie unter Eine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy.

Bereitstellung auf einer Amazon-ECS-Datenverarbeitungsplattform

Dieses Thema enthält Informationen zu den Komponenten und zum Arbeitsablauf von CodeDeploy Bereitstellungen, die die Amazon ECS-Rechenplattform verwenden.

Themen

- Bevor Sie mit einer Amazon-ECS-Bereitstellung beginnen
- Bereitstellungs-Workflow (auf hoher Ebene) auf einer Amazon ECS-Rechenplattform
- Was passiert während einer Amazon ECS-Bereitstellung
- Die Revision Ihrer Bewerbung wird hochgeladen
- Erstellen Sie Ihre Anwendungs- und Bereitstellungsgruppen
- Revision Ihrer Anwendung bereitstellen
- Aktualisierung Ihrer Anwendung
- Bereitstellungen wurden gestoppt und sind fehlgeschlagen
- Neubereitstellungen und Bereitstellungs-Rollbacks
- Amazon blue/green ECS-Bereitstellungen über AWS CloudFormation

Bevor Sie mit einer Amazon-ECS-Bereitstellung beginnen

Bevor Sie mit der Bereitstellung einer Amazon ECS-Anwendung beginnen, müssen Sie Folgendes bereithalten. Einige Anforderungen werden bei der Erstellung Ihrer Bereitstellungsgruppe angegeben, andere sind in der AppSpec Datei angegeben.

Anforderung	Wo angegeben
Amazon-ECS-Cluster	Bereitstellungsgruppe
Amazon-ECS-Service	Bereitstellungsgruppe
Application Load Balancer oder Network Load Balancer	Bereitstellungsgruppe
Produktions-Listener	Bereitstellungsgruppe
Test-Listener (optional)	Bereitstellungsgruppe
Zwei Zielgruppen	Bereitstellungsgruppe
Amazon ECS-Aufgabendefinition	AppSpec datei
Container-Name	AppSpec datei
Container-Port	AppSpec datei

Amazon ECS-Cluster

Ein Amazon-ECS-Cluster ist eine logische Gruppierung von Aufgaben oder Services. Sie geben den Amazon ECS-Cluster an, der Ihren Amazon ECS-Service enthält, wenn Sie die Bereitstellungsgruppe Ihrer CodeDeploy Anwendung erstellen. Weitere Informationen finden Sie unter Amazon ECS-Cluster im Amazon Elastic Container Service-Benutzerhandbuch.

Amazon ECS-Service

Ein Amazon ECS-Service verwaltet und führt bestimmte Instances einer Aufgabendefinition in einem Amazon ECS-Cluster aus. Ihr Amazon ECS-Service muss für aktiviert sein CodeDeploy. Standardmäßig ist ein Amazon ECS-Service für Amazon ECS-Bereitstellungen aktiviert. Wenn Sie Ihre Bereitstellungsgruppe erstellen, entscheiden Sie sich dafür, einen Amazon ECS-Service

bereitzustellen, der sich in Ihrem Amazon ECS-Cluster befindet. Weitere Informationen finden Sie unter Amazon ECS-Services im Amazon Elastic Container Service-Benutzerhandbuch.

Application Load Balancer oder Network Load Balancer

Sie müssen Elastic Load Balancing mit dem Amazon ECS-Service verwenden, den Sie mit einer Amazon ECS-Bereitstellung aktualisieren möchten. Sie können einen Application Load Balancer oder einen Network Load Balancer verwenden. Wir empfehlen einen Application Load Balancer, damit Sie Funktionen wie dynamische Portzuweisung und pfadbasiertes Routing und Prioritätsregeln nutzen können. Sie geben den Load Balancer an, wenn Sie die Bereitstellungsgruppe für Ihre CodeDeploy Anwendung erstellen. Weitere Informationen finden Sie unter Einen Load Balancer, Zielgruppen und Listener für CodeDeploy Amazon ECS-Bereitstellungen einrichten und Creating a Load Balancer im Amazon Elastic Container Service-Benutzerhandbuch.

Ein oder zwei Zuhörer

Ein Listener wird vom Load Balancer verwendet, um Datenverkehr an die Zielgruppen weiterzuleiten. Ein Produktions-Listener ist erforderlich. Sie können optional einen zweiten Test-Listener angeben, der Datenverkehr an den neuen Aufgabensatz weiterleitet, während Sie Validierungstests durchführen. Sie geben einen oder beide Listener an, wenn Sie die Bereitstellungsgruppe erstellen. Wenn Sie die Amazon ECS-Konsole verwenden, um Ihren Amazon ECS-Service zu erstellen, werden Ihre Listener für Sie erstellt. Weitere Informationen finden Sie unter Listener für Ihre Application Load Balancer im Elastic Load Balancing User Guide und Creating a Service im Amazon Elastic Container Service User Guide.

Zwei Amazon ECS-Zielgruppen

Eine Zielgruppe wird verwendet, um Datenverkehr an ein registriertes Ziel weiterzuleiten. Für eine Amazon ECS-Bereitstellung sind zwei Zielgruppen erforderlich: eine für den ursprünglichen Tasksatz Ihrer Amazon ECS-Anwendung und eine für den Ersatz-Tasksatz. CodeDeploy Erstellt während der Bereitstellung einen Ersatzaufgabensatz und leitet den Datenverkehr vom ursprünglichen Aufgabensatz zum neuen um. Sie geben die Zielgruppen an, wenn Sie die Bereitstellungsgruppe der CodeDeploy -Anwendung erstellen.

CodeDeploy Ermittelt während einer Bereitstellung, welche Zielgruppe mit dem Aufgabensatz in Ihrem Amazon ECS-Service verknüpft ist, der den Status hat `PRIMARY` (dies ist der ursprüngliche Aufgabensatz), ordnet ihm eine Zielgruppe zu und ordnet dann die andere Zielgruppe dem Ersatzaufgabensatz zu. Bei der nächsten Bereitstellung wird die Zielgruppe, die dem ursprünglichen Aufgabensatz der jetzigen Bereitstellung zugeordnet ist, dem neuen Aufgabensatz

der nächsten Bereitstellung zugeordnet. Weitere Informationen finden Sie unter Zielgruppen für Ihre Application Load Balancer im Elastic Load Balancing User Guide.

Eine Amazon ECS-Aufgabendefinition

Eine Aufgabendefinition ist erforderlich, um den Docker-Container auszuführen, der Ihre Amazon ECS-Anwendung enthält. Sie geben den ARN Ihrer Aufgabendefinition in der AppSpec Datei Ihrer CodeDeploy Anwendung an. Weitere Informationen finden Sie unter Amazon ECS-Aufgabendefinitionen im Amazon Elastic Container Service-Benutzerhandbuch undAppSpec Abschnitt „Ressourcen“ für Amazon ECS-Bereitstellungen.

Ein Container für Ihre Amazon ECS-Anwendung

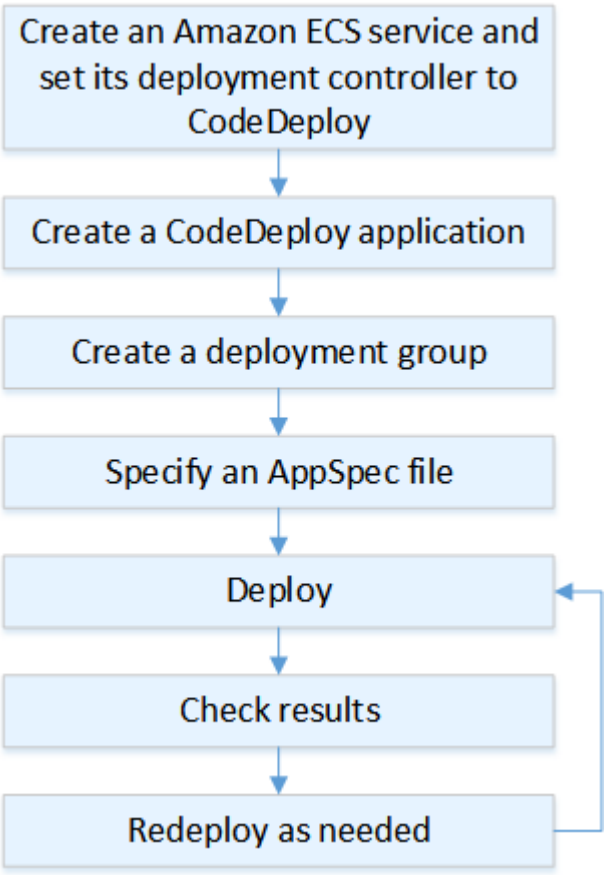
Ein Docker-Container ist eine Software-Einheit, die Code und dessen Abhängigkeiten so verpackt, dass Ihre Anwendung ausgeführt werden kann. Ein Container isoliert die Anwendung so, dass sie in verschiedenen Computing-Umgebungen ausgeführt werden kann. Ihr Load Balancer leitet den Verkehr an einen Container im Taskset Ihrer Amazon ECS-Anwendung weiter. Sie geben den Namen Ihres Containers in der AppSpec Datei Ihrer CodeDeploy Anwendung an. Der in Ihrer AppSpec Datei angegebene Container muss einer der in Ihrer Amazon ECS-Aufgabendefinition angegebenen Container sein. Weitere Informationen finden Sie unter Was ist Amazon Elastic Container Service? im Amazon Elastic Container Service-Benutzerhandbuch undAppSpec Abschnitt „Ressourcen“ für Amazon ECS-Bereitstellungen.

Ein Port für Ihren Ersatz-Aufgabensatz

Während Ihrer Amazon ECS-Bereitstellung leitet Ihr Load Balancer den Datenverkehr an diesen Port auf dem Container weiter, der in der AppSpec Datei Ihrer CodeDeploy Anwendung angegeben ist. Sie geben den Port in der AppSpec Datei Ihrer CodeDeploy Anwendung an. Weitere Informationen finden Sie unter AppSpec Abschnitt „Ressourcen“ für Amazon ECS-Bereitstellungen.

Bereitstellungs-Workflow (auf hoher Ebene) auf einer Amazon ECS-Rechenplattform

Das folgende Diagramm zeigt die wichtigsten Schritte bei der Bereitstellung aktualisierter Amazon ECS-Services.



Zu diesen Schritten gehören:

1. Erstellen Sie eine AWS CodeDeploy Anwendung, indem Sie einen Namen angeben, der eindeutig darstellt, was Sie bereitstellen möchten. Um eine Amazon ECS-Anwendung bereitzustellen, wählen Sie in Ihrer AWS CodeDeploy Anwendung die Amazon ECS-Rechenplattform aus. CodeDeploy verwendet während einer Bereitstellung eine Anwendung, um auf die richtigen Bereitstellungskomponenten wie Bereitstellungsgruppe, Zielgruppen, Listener und das Verhalten bei der Umleitung des Datenverkehrs sowie die Anwendungsversion zu verweisen. Weitere Informationen finden Sie unter Erstellen Sie eine Anwendung mit CodeDeploy.
2. Richten Sie eine Bereitstellungsgruppe ein, indem Sie Folgendes angeben:
   - Name der Bereitstellungsgruppe.
   - Ihr Amazon ECS-Cluster und Ihr Service-Name. Der Deployment Controller des Amazon ECS-Service muss auf eingestellt sein CodeDeploy.
   - Der Produktions-Listener, ein optionaler Test-Listener und Zielgruppen, die während einer Bereitstellung verwendet werden.

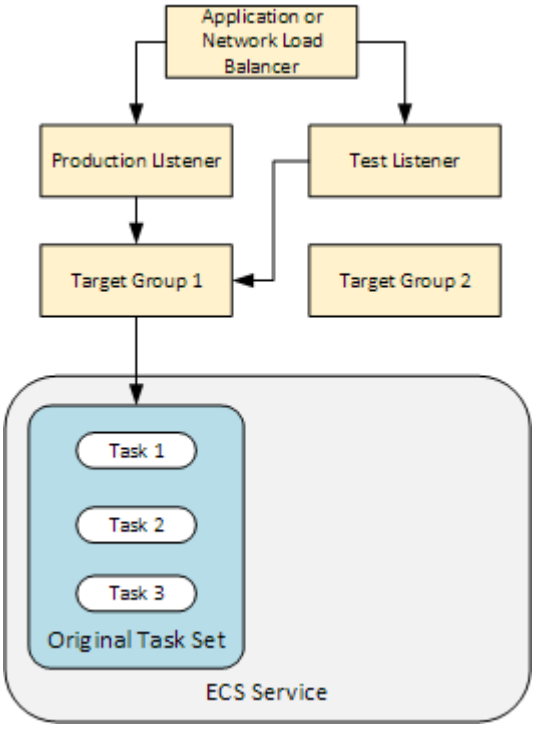
- Bereitstellungseinstellungen, z. B. wann der Produktionsdatenverkehr an den Amazon ECS-Ersatzaufgabensatz in Ihrem Amazon ECS-Service umgeleitet werden soll und wann der ursprüngliche Amazon ECS-Auftragssatz in Ihrem Amazon ECS-Service beendet werden muss.
- Optionale Einstellungen, wie z. B. Auslöser, Alarme und Rollback-Verhalten.

3. Geben Sie eine Anwendungsspezifikationsdatei (AppSpec Datei) an. Sie können es auf Amazon S3 hochladen, es im YAML- oder JSON-Format in die Konsole eingeben oder mit dem SDK AWS CLI oder angeben. Die AppSpec Datei spezifiziert eine Amazon ECS-Aufgabendefinition für die Bereitstellung, eine Container-Namen- und Port-Zuordnung, die zur Weiterleitung des Datenverkehrs verwendet werden, und Lambda-Funktionen, die nach den Hooks des Deployment-Lebenszyklus ausgeführt werden. Der Containername muss ein Container in Ihrer Amazon ECS-Aufgabendefinition sein. Weitere Informationen finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.
4. Stellen Sie die Revision Ihrer Anwendung bereit. AWS CodeDeploy leitet den Verkehr von der ursprünglichen Version eines Task-Sets in Ihrem Amazon ECS-Service zu einem neuen Ersatz-Task-Set um. Zielgruppen werden in der Bereitstellungsgruppe verwendet, um den Datenverkehr an den ursprünglichen und den neuen Aufgabensatz weiterzuleiten. Wenn die Bereitstellung abgeschlossen ist, wird der ursprüngliche Aufgabensatz beendet. Sie können einen optionalen Test-Listener angeben, um Testdatenverkehr an die neue Version weiterzuleiten, bevor der richtige Datenverkehr an sie weitergeleitet wird. Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy.
5. Überprüfen Sie die Bereitstellungsergebnisse. Weitere Informationen finden Sie unter Überwachung von Bereitstellungen in CodeDeploy.

Was passiert während einer Amazon ECS-Bereitstellung

Bevor eine Amazon ECS-Bereitstellung mit einem Test-Listener gestartet wird, müssen Sie dessen Komponenten konfigurieren. Weitere Informationen finden Sie unter Bevor Sie mit einer Amazon-ECS-Bereitstellung beginnen.

Das folgende Diagramm zeigt die Beziehung zwischen diesen Komponenten, wenn eine Amazon ECS-Bereitstellung startbereit ist.



Wenn die Bereitstellung gestartet wird, werden die Ereignisse im Bereitstellungslebenszyklus einzeln ausgeführt. Einige Lebenszyklusereignisse sind Hooks, die nur Lambda-Funktionen ausführen, die in der AppSpec Datei angegeben sind. Die Ereignisse im Bereitstellungslebenszyklus in der folgenden Tabelle sind in der Reihenfolge aufgeführt, in der sie ausgeführt werden. Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“ für eine Amazon ECS-Bereitstellung.

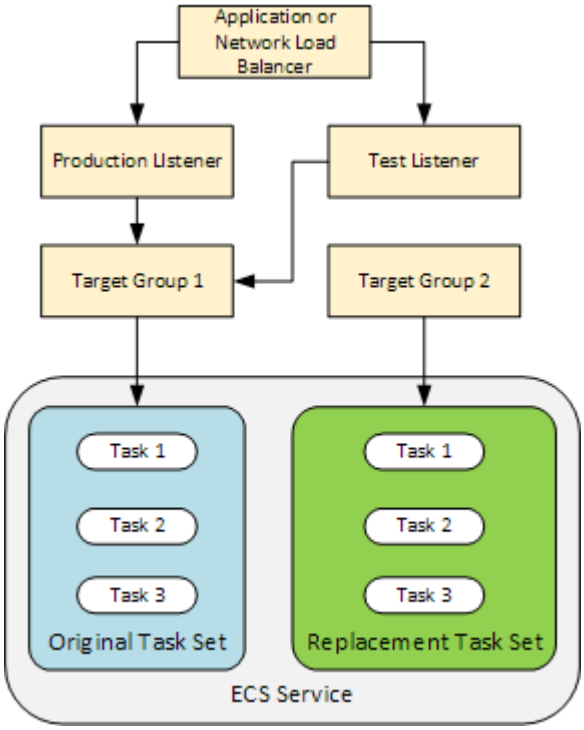
Lebenszyklus-Ereignis	Aktion im Lebenszyklus-Ereignis
`BeforeInstall` (ein Hook für Lambda-Funktionen)	Führen Sie Lambda-Funktionen aus.
Installieren	Richten Sie den Ersatzaufgabensatz ein.
`AfterInstall` (ein Hook für Lambda-Funktionen)	Führen Sie Lambda-Funktionen aus.
AllowTestTraffic	Leiten Sie Datenverkehr aus dem Test-Listener an Zielgruppe 2 weiter.
`AfterAllowTestTraffic` (ein Hook für Lambda-Funktionen)	Führen Sie Lambda-Funktionen aus.

Lebenszyklus-Ereignis	Aktion im Lebenszyklus-Ereignis
`BeforeAllowTraffic` (ein Hook für Lambda-Funktionen)	Führen Sie Lambda-Funktionen aus.
AllowTraffic	Leiten Sie Datenverkehr aus dem Produktions-Listener an Zielgruppe 2 weiter.
`AfterAllowTraffic`	Führen Sie Lambda-Funktionen aus.

> **Note**
>
> Lambda-Funktionen in einem Hook sind optional.

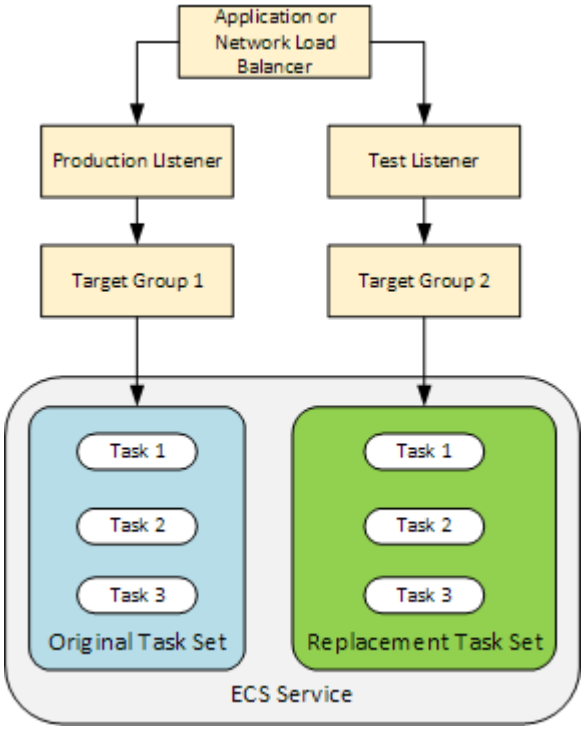
1. Führt alle Lambda-Funktionen aus, die im `BeforeInstall` Hook in der AppSpec Datei angegeben sind.
2. Während des `Install`-Lebenszyklusereignisses:
   a. In Ihrem Amazon ECS-Service wird ein Ersatzaufgabensatz erstellt.
   b. Die aktualisierte containerisierte Anwendung wird im Ersatzaufgabensatz installiert.
   c. Die zweite Zielgruppe wird dem Ersatzaufgabensatz zugeordnet.

   Dieses Diagramm zeigt Bereitstellungskomponenten mit dem neuen Ersatzaufgabensatz. Die containerisierte Anwendung befindet sich in diesem Aufgabensatz. Der Aufgabensatz besteht aus drei Aufgaben. (Eine Anwendung kann eine beliebige Anzahl von Aufgaben umfassen.) Die zweite Zielgruppe wird jetzt dem Ersatzaufgabensatz zugeordnet.



3. Führt alle Lambda-Funktionen aus, die im `AfterInstall` Hook in der AppSpec Datei angegeben sind.

4. Das `AllowTestTraffic`-Ereignis wird aufgerufen. Während dieses Lebenszyklusereignisses leitet der Test-Listener den Datenverkehr an die aktualisierte containerisierte Anwendung weiter.



5. Führt alle Lambda-Funktionen aus, die im `AfterAllowTestTraffic` Hook in der AppSpec Datei angegeben sind. Lambda-Funktionen können die Bereitstellung anhand des Testdatenverkehrs validieren. Beispiel: Eine Lambda-Funktion kann den Datenverkehr an den Test-Listener leiten und Metriken aus dem Ersatzaufgabensatz nachverfolgen. Wenn Rollbacks konfiguriert sind, können Sie einen CloudWatch Alarm konfigurieren, der einen Rollback auslöst, wenn der Validierungstest in Ihrer Lambda-Funktion fehlschlägt.

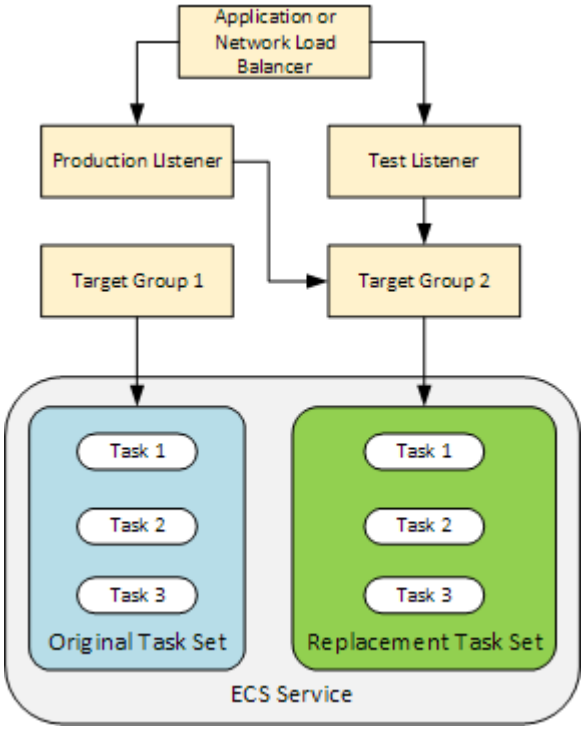
   Nach dem Abschluss der Validierungstests tritt eine der folgenden Situationen ein:

   - Wenn die Validierung fehlschlägt und Rollbacks konfiguriert sind, wird der Bereitstellungsstatus als `Failed` markiert und Komponenten kehren in ihren Zustand zurück, wenn die Bereitstellung gestartet wird.
   - Wenn die Validierung fehlschlägt und Rollbacks nicht konfiguriert sind, wird der Bereitstellungsstatus als `Failed` markiert und Komponenten bleiben in ihrem aktuellen Zustand.
   - Wenn die Validierung erfolgreich ist, wird die Bereitstellung am `BeforeAllowTraffic`-Hook fortgesetzt.

Weitere Informationen finden Sie unter Überwachung von Bereitstellungen mit CloudWatch Alarmen in CodeDeploy, Automatische Rollbacks und Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe.

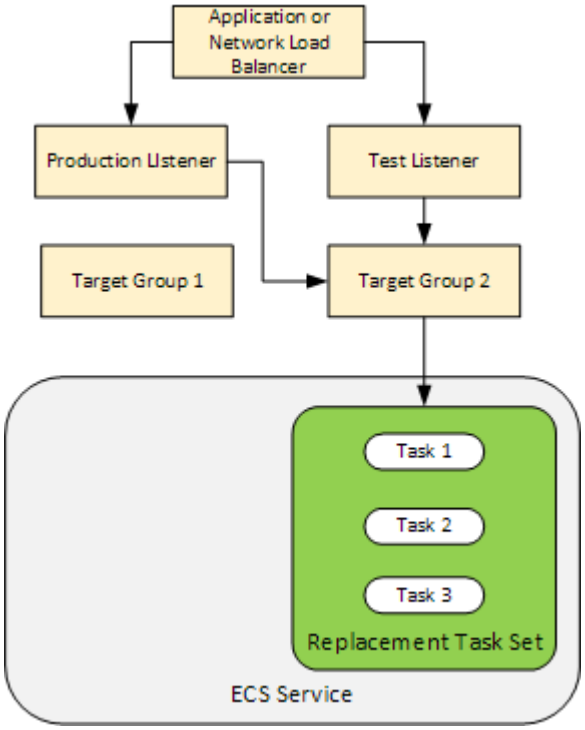
6. Führt alle Lambda-Funktionen aus, die im `BeforeAllowTraffic` Hook in der AppSpec Datei angegeben sind.

7. Das `AllowTraffic`-Ereignis wird aufgerufen. Produktionsdatenverkehr wird vom ursprünglichen Aufgabensatz auf den Ersatzaufgabensatz umgeleitet. Das folgende Diagramm zeigt den Ersatzaufgabensatz, der den Produktionsdatenverkehr empfängt.



8. Führt alle Lambda-Funktionen aus, die im `AfterAllowTraffic` Hook in der AppSpec Datei angegeben sind.

9. Nachdem alle Ereignisse erfolgreich waren, wird der Bereitstellungsstatus auf `Succeeded` gesetzt und die ursprüngliche Aufgabe entfernt.



Die Revision Ihrer Bewerbung wird hochgeladen

Platzieren Sie eine AppSpec Datei in Amazon S3 oder geben Sie sie direkt in die Konsole ein oder AWS CLI. Weitere Informationen finden Sie unter Application Specification Files.

Erstellen Sie Ihre Anwendungs- und Bereitstellungsgruppen

Eine CodeDeploy Bereitstellungsgruppe auf einer Amazon ECS-Rechenplattform identifiziert Listener, die den Datenverkehr an Ihre aktualisierte Amazon ECS-Anwendung weiterleiten sollen, sowie zwei Zielgruppen, die während Ihrer Bereitstellung verwendet wurden. Eine Bereitstellungsgruppe definiert auch eine Reihe von Konfigurationsoptionen, wie z. B. Alarme und Rollback-Konfigurationen.

Revision Ihrer Anwendung bereitstellen

Jetzt sind Sie bereit, den aktualisierten Amazon ECS-Service bereitzustellen, der in Ihrer Bereitstellungsgruppe angegeben ist. Sie können die CodeDeploy Konsole oder den Befehl create-deployment verwenden. Es gibt Parameter, die Sie angeben können, um Ihre Bereitstellung zu steuern, einschließlich der Revision und der Bereitstellungsgruppe.

Aktualisierung Ihrer Anwendung

Sie können Ihre Anwendung aktualisieren und dann die CodeDeploy Konsole verwenden oder den Befehl create-deployment aufrufen, um eine Version zu pushen.

Bereitstellungen wurden gestoppt und sind fehlgeschlagen

Sie können die CodeDeploy Konsole oder den Befehl stop-deployment verwenden, um eine Bereitstellung zu beenden. Wenn Sie versuchen, die Bereitstellung zu stoppen, gibt es drei Möglichkeiten:

- Die Bereitstellung wird beendet, und die Operation gibt den Status "Erfolgreich" zurück. In diesem Fall werden keine Bereitstellungslebenszyklusereignisse mehr für die Bereitstellungsgruppe für die gestoppte Bereitstellung ausgeführt.
- Die Bereitstellung wird nicht sofort beendet, und die Operation gibt den Status "Ausstehend" zurück. In diesem Fall werden einige Bereitstellungslebenszyklusereignisse in der Bereitstellungsgruppe weiterhin ausgeführt. Nachdem die ausstehende Operation abgeschlossen ist, geben nachfolgende Aufrufe zum Stoppen der Bereitstellung den Status "Erfolgreich" zurück.
- Die Bereitstellung kann nicht gestoppt werden, und der Vorgang gibt einen Fehler zurück. Weitere Informationen finden Sie unter Fehlerinformationen und Häufige Fehler in der AWS CodeDeploy API-Referenz.

Neubereitstellungen und Bereitstellungs-Rollbacks

CodeDeploy implementiert Rollbacks, indem der Datenverkehr vom Ersatz-Task-Set zum ursprünglichen Task-Set umgeleitet wird.

Sie können eine Bereitstellungsgruppe für automatisches Rollback konfigurieren, wenn bestimmte Bedingungen erfüllt sind, einschließlich wenn eine Bereitstellung fehlschlägt oder ein Alarm-Überwachungsschwellenwert erreicht ist. Sie können auch die Rollback-Einstellungen für eine Bereitstellungsgruppe in einer einzelnen Bereitstellung überschreiben.

Sie können außerdem ein Rollback einer fehlgeschlagenen Bereitstellung durch erneute manuelle Bereitstellung einer zuvor bereitgestellten Revision durchführen.

In allen Fällen wird der neuen oder rückabgewickelten Bereitstellung eine eigene Bereitstellungs-ID zugewiesen. Die CodeDeploy -Konsole zeigt eine Liste der Bereitstellungen, die das Ergebnis einer automatischen Bereitstellung sind.

Bei einer erneuten Bereitstellung wird die Zielgruppe, die dem ursprünglichen Aufgabensatz der jetzigen Bereitstellung zugeordnet ist, dem neuen Aufgabensatz der erneuten Bereitstellung zugeordnet.

Weitere Informationen finden Sie unter Eine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy.

Amazon blue/green ECS-Bereitstellungen über AWS CloudFormation

Sie können es verwenden AWS CloudFormation , um Amazon blue/green ECS-Bereitstellungen über CodeDeploy zu verwalten. Weitere Informationen finden Sie unter Erstellen Sie eine Amazon blue/green ECS-Bereitstellung über CloudFormation.

> **Note**
>
> Die Verwaltung von Amazon blue/green ECS-Bereitstellungen mit CloudFormation ist in der Region Asien-Pazifik (Osaka) nicht verfügbar.

Bereitstellungen auf einer Rechenplattform EC2/On-Premises

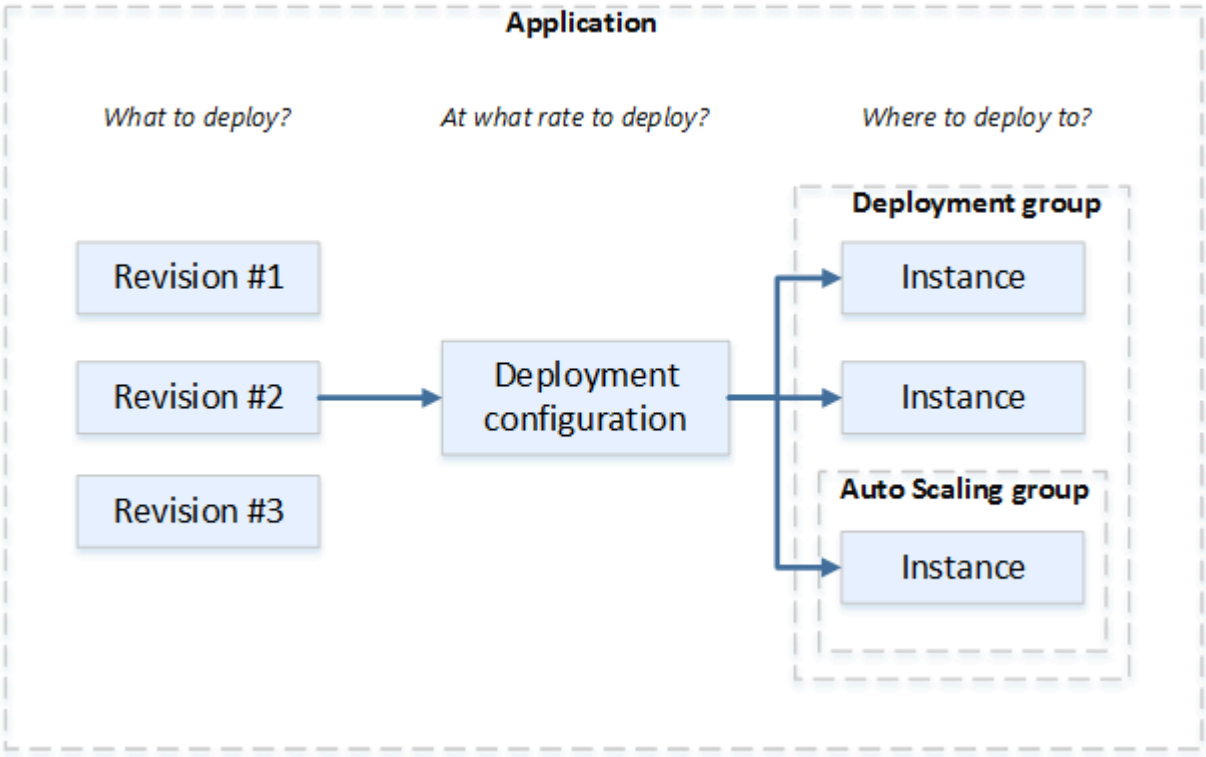
Dieses Thema enthält Informationen zu den Komponenten und zum Arbeitsablauf von CodeDeploy Bereitstellungen, die die EC2/On-Premises Rechenplattform verwenden. Informationen zu blue/green Bereitstellungen finden Sie unter. Überblick über eine Bereitstellung blue/green

Themen

- Bereitstellungskomponenten auf einer EC2/on-premises Rechenplattform
- Bereitstellungs-Workflow auf einer EC2/on-premises Rechenplattform
- Instances einrichten
- Die Revision Ihrer Anwendung wird hochgeladen
- Erstellen Sie Ihre Anwendungs- und Bereitstellungsgruppen
- Revision Ihrer Anwendung bereitstellen
- Aktualisierung Ihrer Anwendung
- Bereitstellungen wurden gestoppt und sind fehlgeschlagen
- Neubereitstellungen und Bereitstellungs-Rollbacks

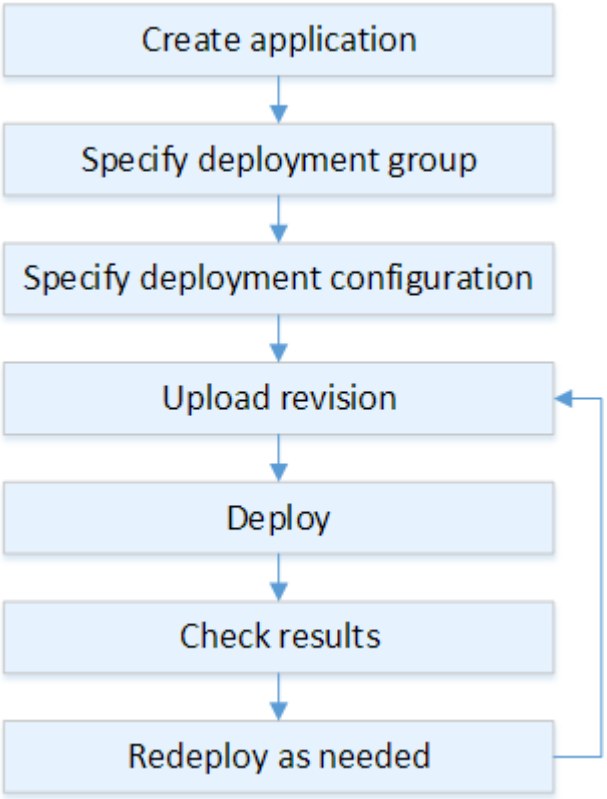
Bereitstellungskomponenten auf einer EC2/on-premises Rechenplattform

Das folgende Diagramm zeigt die Komponenten einer CodeDeploy Bereitstellung auf einer EC2/On-Premises Rechenplattform.



Bereitstellungs-Workflow auf einer EC2/on-premises Rechenplattform

Das folgende Diagramm zeigt die wichtigsten Schritte in der Bereitstellung von Anwendungsrevisionen:



Zu diesen Schritten gehören:

1. Erstellen Sie eine Anwendung und geben Sie ihr einen Namen, der die Anwendungsrevisionen, die Sie bereitstellen möchten, und die Rechenplattform für Ihre Anwendung eindeutig identifiziert. CodeDeploy verwendet diesen Namen während einer Bereitstellung, um sicherzustellen, dass er auf die richtigen Bereitstellungskomponenten verweist, z. B. die Bereitstellungsgruppe, die Bereitstellungskonfiguration und die Anwendungsversion. Weitere Informationen finden Sie unter Erstellen Sie eine Anwendung mit CodeDeploy.
2. Richten Sie eine Bereitstellungsgruppe ein, indem Sie einen Bereitstellungstyp und die Instances angeben, für die Sie die Anwendungsrevisionen bereitstellen möchten. Eine In-Situ-Bereitstellung aktualisiert Instances mit der neuesten Anwendungsrevision. Bei einer blue/green Bereitstellung wird ein Ersatzsatz von Instances für die Bereitstellungsgruppe mit einem Load Balancer registriert und die Registrierung der ursprünglichen Instances aufgehoben.

   Sie können die auf die Instances angewendeten Tags, die Amazon EC2 Auto Scaling Scaling-Gruppennamen oder beides angeben.

   Wenn Sie eine Gruppe von Tags in einer Bereitstellungsgruppe angeben, erfolgt die Bereitstellung CodeDeploy auf Instances, auf die mindestens eines der angegebenen Tags angewendet wurde.

Wenn Sie zwei oder mehr Tag-Gruppen angeben, erfolgt CodeDeploy die Bereitstellung nur für die Instances, die die Kriterien für jede der Tag-Gruppen erfüllen. Weitere Informationen finden Sie unter Tagging Instances for Deployments.

In allen Fällen müssen die Instances für die Verwendung in einer Bereitstellung konfiguriert sein (d. h. sie müssen markiert sein oder zu einer Amazon EC2 Auto Scaling Scaling-Gruppe gehören) und der CodeDeploy Agent muss installiert sein und ausgeführt werden.

Sie können eine CloudFormation Vorlage verwenden, um schnell eine Amazon EC2 EC2-Instance auf Basis von Amazon Linux oder Windows Server einzurichten. Wir stellen Ihnen auch den eigenständigen CodeDeploy Agenten zur Verfügung, sodass Sie ihn auf Amazon Linux-, Ubuntu Server-, Red Hat Enterprise Linux (RHEL) - oder Windows Server-Instances installieren können. Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy.

Sie können auch die folgenden Optionen angeben:

- Amazon SNS SNS-Benachrichtigungen. Erstellen Sie Auslöser, die Benachrichtigungen an Abonnenten eines Amazon SNS SNS-Themas senden, wenn bestimmte Ereignisse, wie Erfolgs- oder Fehlschlagsereignisse, in Bereitstellungen und Instances auftreten. Weitere Informationen finden Sie unter Monitoring Deployments with Amazon SNS Event Notifications.
- Alarm-based Bereitstellungsmanagement. Implementieren Sie Amazon CloudWatch Alarm Monitoring, um Bereitstellungen zu stoppen, wenn Ihre Messwerte die festgelegten Schwellenwerte über- oder unterschreiten. CloudWatch
- Automatische Bereitstellungsrollbacks. Konfigurieren Sie das automatische Zurücksetzen einer Bereitstellung auf die letzte funktionierende Revision, wenn eine Bereitstellung fehlschlägt oder ein Alarmschwellenwert erreicht ist.

3. Legen Sie eine Bereitstellungskonfiguration fest, um anzugeben, wie viele Instances Ihrer Anwendungsrevisionen gleichzeitig bereitgestellt werden sollen, und um die Bedingungen für Erfolg und Fehler für die Bereitstellung zu beschreiben. Weitere Informationen finden Sie unter View Deployment Configuration Details.
4. Laden Sie eine Anwendungsrevision auf Amazon S3 hoch oder GitHub. Zusätzlich zu den Dateien, die Sie bereitstellen möchten, und allen Skripten, die Sie während der Bereitstellung ausführen möchten, müssen Sie eine Anwendungsspezifikationsdatei (AppSpec Datei) beifügen. Diese Datei enthält Bereitstellungsanweisungen, wie zum Beispiel wohin die Dateien auf jeder Instance kopiert werden und wann die Bereitstellungsskripts ausgeführt werden sollen. Weitere Informationen finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.

5. Stellen Sie Ihre Anwendungsrevision für die Bereitstellungsgruppe bereit. Der CodeDeploy Agent auf jeder Instance in der Bereitstellungsgruppe kopiert Ihre Anwendungsversion von Amazon S3 oder GitHub auf die Instance. Der CodeDeploy Agent entbündelt dann die Revision und kopiert die Dateien anhand der AppSpec Datei an die angegebenen Speicherorte und führt alle Bereitstellungsskripte aus. Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy.
6. Überprüfen Sie die Bereitstellungsergebnisse. Weitere Informationen finden Sie unter Überwachung von Bereitstellungen in CodeDeploy.
7. Stellen Sie eine Revision erneut bereit. Dies ist sinnvoll, wenn Sie einen Fehler im Quelleinhalt korrigieren müssen oder die Bereitstellungsskripts in einer anderen Reihenfolge ausführen möchten oder eine Bereitstellung fehlgeschlagen ist. Bündeln Sie dazu Ihren überarbeiteten Quellinhalt, alle Bereitstellungsskripten und die AppSpec Datei in eine neue Version und laden Sie die Version dann in den Amazon S3-Bucket oder das Amazon GitHub S3-Repository hoch. Anschließend führen Sie eine neue Bereitstellung für dieselbe Bereitstellungsgruppe mit der neuen Revision aus. Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy.

Instances einrichten

Sie müssen Instances einrichten, bevor Sie erstmalig Anwendungsrevisionen bereitstellen können. Wenn eine Anwendungsrevision drei Produktionsserver und zwei Backup-Server erfordert, starten oder verwenden Sie fünf Instances.

So stellen Sie Instances manuell bereit:

1. Installieren Sie den CodeDeploy Agenten auf den Instanzen. Der CodeDeploy Agent kann auf Amazon Linux-, Ubuntu Server-, RHEL- und Windows Server-Instances installiert werden.
2. Aktivieren Sie das Tagging, wenn Sie Tags verwenden, um Instances in einer Bereitstellungsgruppe zu identifizieren. CodeDeploy verwendet Tags, um Instanzen zu identifizieren und in CodeDeploy Bereitstellungsgruppen zu gruppieren. Im Tutorial "Erste Schritte" wurde zwar beides verwendet, aber Sie können einfach einen Schlüssel oder einen Wert verwenden, um ein Tag für eine Bereitstellungsgruppe zu definieren.
3. Starten Sie Amazon EC2 EC2-Instances mit angehängtem IAM-Instance-Profil. Das IAM-Instance-Profil muss beim Start an eine Amazon EC2 EC2-Instance angehängt werden, damit der CodeDeploy Agent die Identität der Instance überprüfen kann.

4. Erstellen Sie eine Servicerolle. Stellen Sie Servicezugriff bereit, damit Sie die Tags in Ihrem AWS Konto erweitern CodeDeploy können.

Bei einer ersten Bereitstellung erledigt die CloudFormation Vorlage all dies für Sie. Es erstellt und konfiguriert neue, einzelne Amazon EC2 EC2-Instances auf Basis von Amazon Linux oder Windows Server, wobei der CodeDeploy Agent bereits installiert ist. Weitere Informationen finden Sie unter Arbeiten mit Instanzen für CodeDeploy.

> **Note**
>
> Für eine blue/green Bereitstellung können Sie wählen, ob Sie bereits vorhandene Instances für die Ersatzumgebung verwenden oder sich im Rahmen des Bereitstellungsprozesses neue Instances CodeDeploy bereitstellen lassen möchten.

Die Revision Ihrer Anwendung wird hochgeladen

Platzieren Sie eine AppSpec Datei im Stammordner der Ordnerstruktur für den Quellinhalt Ihrer Anwendung. Weitere Informationen finden Sie unter Application Specification Files.

Verpacken Sie die Quellinhalts-Ordnerstruktur der Anwendung in ein Archivdateiformat, beispielsweise ZIP, TAR oder komprimiertes TAR. Laden Sie die Archivdatei (die Revision) in einen Amazon S3-Bucket oder ein Amazon GitHub S3-Repository hoch.

> **Note**
>
> Die Tar- und komprimierten Tar-Archivdateiformate (.tar und .tar.gz) werden für Windows Server-Instances nicht unterstützt.

Erstellen Sie Ihre Anwendungs- und Bereitstellungsgruppen

Eine CodeDeploy Bereitstellungsgruppe identifiziert eine Sammlung von Instances anhand ihrer Tags, Amazon EC2 Auto Scaling Scaling-Gruppennamen oder beidem. Es können mehrere Anwendungsrevisionen für dieselbe Instance bereitgestellt werden. Eine Anwendungsrevision kann für mehrere Instances bereitgestellt werden.

Sie können beispielsweise das Tag "Prod" zu den drei Produktionsservern und "Backup" zu den beiden Backup-Servern hinzufügen. Diese beiden Tags können verwendet werden, um zwei

verschiedene Bereitstellungsgruppen in der CodeDeploy Anwendung zu erstellen, sodass Sie wählen können, welcher Serversatz (oder beide) an einer Bereitstellung teilnehmen soll.

Sie können mehrere Tag-Gruppen in einer Bereitstellungsgruppe verwenden, um Bereitstellungen auf eine kleinere Auswahl von Instances einzuschränken. Weitere Informationen finden Sie unter Tagging Instances for Deployments.

Revision Ihrer Anwendung bereitstellen

Jetzt sind Sie bereit, Ihre Anwendungsrevision von Amazon S3 oder GitHub in der Bereitstellungsgruppe bereitzustellen. Sie können die CodeDeploy Konsole oder den Befehl create-deployment verwenden. Es gibt Parameter, die Sie angeben können, um Ihre Bereitstellung zu steuern, einschließlich der Revision, Bereitstellungsgruppe und Bereitstellungskonfiguration.

Aktualisierung Ihrer Anwendung

Sie können Ihre Anwendung aktualisieren und dann die CodeDeploy Konsole verwenden oder den Befehl create-deployment aufrufen, um eine Version zu pushen.

Bereitstellungen wurden gestoppt und sind fehlgeschlagen

Sie können die CodeDeploy Konsole oder den Befehl stop-deployment verwenden, um eine Bereitstellung zu beenden. Wenn Sie versuchen, die Bereitstellung zu stoppen, gibt es drei Möglichkeiten:

- Die Bereitstellung wird beendet, und die Operation gibt den Status "Erfolgreich" zurück. In diesem Fall werden keine Bereitstellungslebenszyklusereignisse mehr für die Bereitstellungsgruppe für die gestoppte Bereitstellung ausgeführt. Einige Dateien wurden möglicherweise bereits kopiert und einige Skripte bereits für eine oder mehrere der Instances in der Bereitstellungsgruppe ausgeführt.
- Die Bereitstellung wird nicht sofort beendet, und die Operation gibt den Status "Ausstehend" zurück. In diesem Fall werden einige Bereitstellungslebenszyklusereignisse in der Bereitstellungsgruppe weiterhin ausgeführt. Einige Dateien wurden möglicherweise bereits kopiert und einige Skripte bereits für eine oder mehrere der Instances in der Bereitstellungsgruppe ausgeführt. Nachdem die ausstehende Operation abgeschlossen ist, geben nachfolgende Aufrufe zum Stoppen der Bereitstellung den Status "Erfolgreich" zurück.
- Die Bereitstellung kann nicht gestoppt werden, und der Vorgang gibt einen Fehler zurück. Weitere Informationen finden Sie unter ErrorInformationHäufige Fehler in der AWS CodeDeploy API-Referenz.

Genauso wie gestoppte Bereitstellungen können fehlgeschlagene Bereitstellungen dazu führen, dass einige Ereignisse im Bereitstellungslebenszyklus bereits für eine oder mehrere Instances in der Bereitstellungsgruppe ausgeführt wurden. Um herauszufinden, warum eine Bereitstellung fehlgeschlagen ist, können Sie die CodeDeploy Konsole verwenden, den Befehl get-deployment-instance aufrufen oder die Protokolldateidaten der fehlgeschlagenen Bereitstellung analysieren. Weitere Informationen erhalten Sie unter Revision der Anwendung und Bereinigung der Protokolldateien und Protokolldaten für CodeDeploy EC2/On-Premises Bereitstellungen anzeigen.

Neubereitstellungen und Bereitstellungs-Rollbacks

CodeDeploy implementiert Rollbacks, indem eine zuvor bereitgestellte Version als neue Bereitstellung erneut bereitgestellt wird.

Sie können eine Bereitstellungsgruppe für automatisches Rollback konfigurieren, wenn bestimmte Bedingungen erfüllt sind, einschließlich wenn eine Bereitstellung fehlschlägt oder ein Alarm-Überwachungsschwellenwert erreicht ist. Sie können auch die Rollback-Einstellungen für eine Bereitstellungsgruppe in einer einzelnen Bereitstellung überschreiben.

Sie können außerdem ein Rollback einer fehlgeschlagenen Bereitstellung durch erneute manuelle Bereitstellung einer zuvor bereitgestellten Revision durchführen.

In allen Fällen wird der neuen oder rückabgewickelten Bereitstellung eine eigene Bereitstellungs-ID zugewiesen. Die Liste der Bereitstellungen, die Sie in der CodeDeploy Konsole einsehen können, zeigt, welche das Ergebnis einer automatischen Bereitstellung sind.

Weitere Informationen finden Sie unter Eine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy.

CodeDeploy Anwendungsspezifikationsdateien (AppSpec)

Eine Anwendungsspezifikationsdatei (AppSpec Datei), die nur für gilt CodeDeploy, ist eine Datei im YAML - oder JSON-Format. Die AppSpec Datei wird verwendet, um jede Bereitstellung als eine Reihe von Lifecycle-Event-Hooks zu verwalten, die in der Datei definiert sind.

Hinweise zum Erstellen einer wohlgeformten AppSpec Datei finden Sie unterCodeDeploy AppSpec Dateiverweis.

Themen

- AppSpec Dateien auf einer Amazon ECS Compute Platform
- AppSpec Dateien auf einem AWS Lambda Plattform für die Datenverarbeitung

- AppSpec Dateien auf einer EC2/on-premises Rechenplattform
- Wie der CodeDeploy Agent die AppSpec Datei verwendet

AppSpec Dateien auf einer Amazon ECS Compute Platform

Wenn Ihre Anwendung die Amazon ECS-Rechenplattform verwendet, kann die AppSpec Datei entweder mit YAML oder JSON formatiert werden. Sie kann auch direkt in den Editor der Konsole eingegeben werden. Die AppSpec Datei wird verwendet, um Folgendes anzugeben:

- Der Name des Amazon ECS-Service sowie der Name und der Port des Containers, die für die Weiterleitung des Datenverkehrs an den neuen Tasksatz verwendet werden.
- Die Funktionen können als Validierungstests verwendet werden.

Sie können Lambda-Validierungsfunktionen nach Ereignissen im Bereitstellungslebenszyklus ausführen. Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“ für eine Amazon ECS-Bereitstellung, AppSpec Dateistruktur für Amazon ECS-Bereitstellungen und AppSpec Dateibeispiel für eine Amazon ECS-Bereitstellung.

AppSpec Dateien auf einem AWS Lambda Plattform für die Datenverarbeitung

Wenn Ihre Anwendung die AWS Lambda-Rechenplattform verwendet, kann die AppSpec Datei entweder mit YAML oder JSON formatiert werden. Sie kann auch direkt in den Editor der Konsole eingegeben werden. Die AppSpec Datei wird verwendet, um Folgendes anzugeben:

- Die Version der AWS Lambda Funktion, die bereitgestellt werden soll.
- Die Funktionen können als Validierungstests verwendet werden.

Sie können Lambda-Validierungsfunktionen nach Ereignissen im Bereitstellungslebenszyklus ausführen. Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“ für einen AWS Lambda-Bereitstellung.

AppSpec Dateien auf einer EC2/on-premises Rechenplattform

Wenn Ihre Anwendung die EC2/On-Premises Rechenplattform verwendet, ist die AppSpec Datei immer vorhanden YAML-formatted. Die AppSpec Datei wird verwendet, um:

- Ordnen Sie die Quelldateien in Ihrer Anwendungsrevision den entsprechenden Zielen auf der Instance zu.
- Geben Sie benutzerdefinierte Berechtigungen für bereitgestellte Dateien an.
- Legen Sie Scripts fest, die auf jeder Instance in verschiedenen Phasen des Bereitstellungsprozesses ausgeführt werden.

Sie können nach vielen einzelnen Ereignissen im Bereitstellungslebenszyklus Skripts auf einer Instanz ausführen. CodeDeploy führt nur die in der Datei angegebenen Skripts aus, aber diese Skripts können andere Skripts auf der Instance aufrufen. Sie können alle Arten von Scripts ausführen, sofern diese vom Betriebssystem auf der jeweiligen Instance unterstützt werden. Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“ für eine Bereitstellung EC2/On-Premises.

Wie der CodeDeploy Agent die AppSpec Datei verwendet

Während der Bereitstellung sucht der CodeDeploy Agent im Abschnitt Hooks der AppSpec Datei nach dem Namen des aktuellen Ereignisses. Wenn das Ereignis nicht gefunden wird, fährt der CodeDeploy -Agent mit dem nächsten Schritt fort. Wenn das Ereignis gefunden wird, ruft der CodeDeploy Agent die Liste der auszuführenden Skripts ab. Die Scripts werden sequenziell ausgeführt werden, und zwar in der Reihenfolge, in der sie in der Datei aufgeführt sind. Der Status jedes Skripts wird in der CodeDeploy Agent-Protokolldatei auf der Instanz protokolliert.

Wenn ein Script erfolgreich ausgeführt wird, wird der Beendigungscode 0 (Null) ausgegeben.

> **Note**
>
> Der CodeDeploy Agent wird nicht in einer AWS Lambda- oder Amazon ECS-Bereitstellung verwendet.

Während des Installationsereignisses bestimmt der CodeDeploy Agent anhand der im Dateiabschnitt der AppSpec Datei definierten Zuordnungen, welche Ordner oder Dateien von der Revision in die Instance kopiert werden sollen.

Wenn der auf dem Betriebssystem installierte CodeDeploy Agent nicht mit den Angaben in der AppSpec Datei übereinstimmt, schlägt die Bereitstellung fehl.

Hinweise zu CodeDeploy Agent-Protokolldateien finden Sie unterMit dem CodeDeploy Agenten arbeiten.

Erste Schritte mit CodeDeploy

Themen

- Schritt 1: Einrichtung
- Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy
- Schritt 3: Beschränken Sie die CodeDeploy Benutzerberechtigungen
- Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances

Schritt 1: Einrichtung

Vor der ersten Verwendung AWS CodeDeploy müssen Sie die Einrichtungsschritte abschließen. Die Schritte umfassen die Erstellung eines AWS Kontos (falls Sie noch keines haben) und eines Administratorbenutzers.

In diesem Handbuch wird der Administratorbenutzer als CodeDeployAdministratorbenutzer bezeichnet.

Melden Sie sich an für eine AWS-Konto

Um loszulegen AWS, benötigen Sie eine AWS-Konto. Es gibt zwei Möglichkeiten, sich anzumelden AWS. Wenn Sie die vollständige Kontrolle über die Kontokonfiguration und den Speicherort Ihrer Inhalte haben möchten oder wenn Sie regulierte Workloads wie HIPAA oder FedRAMP haben, verwenden Sie Sign up for (advanced). AWS Wenn Sie ein neuer Benutzer sind und einen Anmeldenamen verwenden möchten, den Sie bereits besitzen, um sich anzumelden AWS und eine AWS Umgebung mit vorkonfigurierten Standardeinstellungen in select zu erstellen, verwenden Sie Sign up for. AWS-RegionenAWS

> **Important**
>
> Wir empfehlen Ihnen dringend, den CodeDeploy administrativen Benutzer als Workforce Identity (ein im IAM Identity Center verwalteter Benutzer) mit dem zu konfigurieren. AWS CLI Bei vielen der Verfahren in diesem Handbuch wird davon ausgegangen, dass Sie den AWS CLI zur Durchführung von Konfigurationen verwenden.

> **Important**
>
> Wenn Sie das konfigurieren AWS CLI, werden Sie möglicherweise aufgefordert, eine AWS Region anzugeben. Wählen Sie eine der unterstützten Regionen aus, die unter Region und Endpunkte in der Allgemeine AWS-Referenzaufgeführt sind.

Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy

In werden Servicerollen verwendet AWS, um einem AWS Service Berechtigungen zu erteilen, sodass dieser auf Ressourcen zugreifen AWS kann. Die Richtlinien, die Sie der Servicerolle anfügen, bestimmen, auf welche -Ressourcen der Service zugreifen und was er mit diesen Ressourcen tun kann.

Der Servicerolle, für die Sie sie erstellen, CodeDeploy müssen die für Ihre Rechenplattform erforderlichen Berechtigungen erteilt werden. Wenn Sie die Bereitstellung auf mehr als einer Rechenplattform durchführen, erstellen Sie für jede Plattform eine Servicerolle. Um Berechtigungen hinzuzufügen, fügen Sie eine oder mehrere der folgenden AWS bereitgestellten Richtlinien hinzu:

Fügen Sie für EC2/On-Premises Bereitstellungen die **`AWSCodeDeployRole`** Richtlinie bei. Sie stellt die Berechtigungen für Ihre Servicerolle bereit:

- Lesen Sie die Tags auf Ihren Instances oder identifizieren Sie Ihre Amazon EC2-Instances anhand der Amazon EC2 Auto Scaling-Gruppennamen.
- Lesen, erstellen, aktualisieren und löschen Sie Amazon EC2 Auto Scaling Scaling-Gruppen, Lifecycle-Hooks und Skalierungsrichtlinien.
- Veröffentlichen Sie Informationen zu Amazon SNS SNS-Themen.
- Rufen Sie Informationen über CloudWatch Alarme ab.
- Lesen und aktualisieren Sie Elastic Load Balancing.

> **Note**
>
> Wenn Sie Ihre Auto Scaling Scaling-Gruppe mit einer Startvorlage erstellen, müssen Sie die folgenden Berechtigungen hinzufügen:
>
> - `ec2:RunInstances`
> - `ec2:CreateTags`

- `iam:PassRole`

Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle Erstellen einer Startvorlage für eine Auto Scaling Scaling-Gruppe und Unterstützung von Startvorlagen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Wenn Sie für Amazon ECS-Bereitstellungen vollen Zugriff auf die Support-Services wünschen, fügen Sie die **`AWSCodeDeployRoleForECS`** Richtlinie bei. Sie stellt die Berechtigungen für Ihre Servicerolle bereit:

- Lesen, aktualisieren und löschen Sie Amazon ECS-Aufgabensätze.
- Aktualisieren Sie die Zielgruppen, Listener und Regeln von Elastic Load Balancing.
- Rufen Sie Funktionen auf AWS Lambda .
- Greifen Sie auf Revisionsdateien in Amazon S3 S3-Buckets zu.
- Rufen Sie Informationen über CloudWatch Alarme ab.
- Veröffentlichen Sie Informationen zu Amazon SNS SNS-Themen.

Wenn Sie für Amazon ECS-Bereitstellungen eingeschränkten Zugriff auf Support-Services wünschen, fügen Sie die **`AWSCodeDeployRoleForECSLimited`** Richtlinie bei. Sie stellt die Berechtigungen für Ihre Servicerolle bereit:

- Lesen, aktualisieren und löschen Sie Amazon ECS-Aufgabensätze.
- Rufen Sie Informationen über CloudWatch Alarme ab.
- Veröffentlichen Sie Informationen zu Amazon SNS SNS-Themen.

Wenn Sie für AWS Lambda-Bereitstellungen die Veröffentlichung auf Amazon SNS zulassen möchten, fügen Sie die Richtlinie bei. **`AWSCodeDeployRoleForLambda`** Sie stellt die Berechtigungen für Ihre Servicerolle bereit:

- Funktionen und Aliase lesen, aktualisieren und aufrufen AWS Lambda .
- Greifen Sie auf Revisionsdateien in Amazon S3 S3-Buckets zu.
- Rufen Sie Informationen über CloudWatch Alarme ab.
- Veröffentlichen Sie Informationen zu Amazon SNS SNS-Themen.

Wenn Sie für AWS Lambda-Bereitstellungen den Zugriff auf Amazon SNS einschränken möchten, fügen Sie die Richtlinie bei. **`AWSCodeDeployRoleForLambdaLimited`** Sie stellt die Berechtigungen für Ihre Servicerolle bereit:

- Funktionen und Aliase lesen, aktualisieren und aufrufen AWS Lambda .
- Greifen Sie auf Revisionsdateien in Amazon S3 S3-Buckets zu.
- Rufen Sie Informationen über CloudWatch Alarme ab.

Im Rahmen der Einrichtung dieser Servicerolle aktualisieren Sie auch die Vertrauensbeziehung mit den Endpunkten, denen Zugriff gewährt werden soll.

Sie können eine Servicerolle mit der IAM-Konsole AWS CLI, den oder den IAM-APIs erstellen.

Themen

- Erstellen Sie eine Servicerolle (Konsole)
- Erstellen Sie eine Servicerolle (CLI)
- Rufen Sie die Servicerolle ARN ab (Konsole)
- Rufen Sie die Servicerolle ARN (CLI) ab

Erstellen Sie eine Servicerolle (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
3. Wählen Sie AWS Service und dann in der Dropdownliste unter Anwendungsfall die Option. CodeDeploy
4. Wählen Sie Ihren Anwendungsfall aus:
   - Wählen Sie CodeDeployfür EC2/On-Premises Bereitstellungen.
   - Wählen Sie CodeDeploy für AWS Lambda-Bereitstellungen Lambda.
   - Wählen Sie für Amazon ECS-Bereitstellungen CodeDeploy — ECS.
5. Wählen Sie Weiter aus.
6. Auf der Seite „Berechtigungen hinzufügen“ wird die richtige Berechtigungsrichtlinie für den Anwendungsfall angezeigt. Wählen Sie Weiter aus.

7. Geben Sie auf der Seite Name, Überprüfung und Erstellung im Feld Rollenname einen Namen für die Servicerolle ein (z. B.**`CodeDeployServiceRole`**), und wählen Sie dann Rolle erstellen aus.

   Sie können unter Rollenbeschreibung auch eine Beschreibung für diese Servicerolle eingeben.

8. Wenn diese Servicerolle zum Zugriff auf alle derzeit unterstützten Endpunkte berechtigt sein soll, sind Sie mit diesem Verfahren fertig.

   Um den Zugriff dieser Servicerolle auf einige Endpunkte zu beschränken, fahren Sie mit den verbleibenden Schritten in diesem Verfahren fort.

9. Suchen Sie in der Rollenliste nach der Rolle, die Sie gerade erstellt haben, und wählen Sie sie aus (`CodeDeployServiceRole`).

10. Wählen Sie die Registerkarte Trust relationships (Vertrauensstellungen).

11. Wählen Sie Vertrauensrichtlinie bearbeiten aus.

    Sie sollten die folgende Richtlinie sehen, die der Servicerolle die Berechtigung zum Zugriff auf alle unterstützten Endpunkte gibt:

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

    Um der Servicerolle nur einigen unterstützten Endpunkten Zugriff zu gewähren, ersetzen Sie den Inhalt des Textfeldes für die Vertrauensrichtlinie durch die folgende Richtlinie. Entfernen Sie die Zeilen für die Endpoints, auf die Sie den Zugriff verhindern möchten, und wählen Sie dann Richtlinie aktualisieren aus.

```
{
    "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"
        }
    ]
```

```
}
```

Weitere Informationen zum Erstellen von Servicerollen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen für einen AWS Dienst im IAM-Benutzerhandbuch.

Erstellen Sie eine Servicerolle (CLI)

1. Erstellen Sie auf Ihrem Entwicklungscomputer eine Textdatei, beispielsweise mit dem Namen `CodeDeployDemo-Trust.json`. Mit dieser Datei wird CodeDeploy erlaubt, in Ihrem Namen zu agieren.

   Führen Sie eine der folgenden Aktionen aus:

   - Um Zugriff auf alle unterstützten AWS Regionen zu gewähren, speichern Sie den folgenden Inhalt in der Datei:

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

   - Um nur einigen unterstützten Regionen Zugriff zu gewähren, geben Sie den folgenden Inhalt in die Datei ein und entfernen Sie die Zeilen für die Regionen, für die Sie den Zugriff ausschließen möchten:

     ```
     {
         "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**
>
> Geben Sie nach dem letzten Endpunkt der Liste kein Komma ein.

2. Rufen Sie aus dem gleichen Verzeichnis den Befehl create-role auf, um eine Servicerolle namens **`CodeDeployServiceRole`** unter Verwendung der Informationen zu erstellen, die in der gerade von Ihnen erstellten Textdatei enthalten sind:

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

> **Important**
>
> Achten Sie darauf, dass `file://` vor dem Dateinamen steht. Dies ist bei diesem Befehl erforderlich.

Notieren Sie sich den Wert des Eintrags `Arn` unter dem Objekt `Role` in der Befehlsausgabe. Sie brauchen ihn später beim Erstellen von Bereitstellungsgruppen. Wenn Sie den Wert vergessen, befolgen Sie die Anweisungen in Rufen Sie die Servicerolle ARN (CLI) ab.

3. Welche verwaltete Richtlinie Sie verwenden, hängt von der Rechenplattform ab.

   - Wenn Ihre Bereitstellung auf einer EC2/On-Premises Rechenplattform erfolgt:

     Rufen Sie den attach-role-policy Befehl auf, um der Servicerolle mit dem Namen **`CodeDeployServiceRole`** die Berechtigungen zu erteilen, die auf der genannten **`AWSCodeDeployRole`** IAM-verwalteten Richtlinie basieren. Beispiel:

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

   - Wenn Ihre Bereitstellung auf einer AWS Lambda-Rechenplattform erfolgt:

     Rufen Sie den attach-role-policy Befehl auf, um der Servicerolle mit dem Namen **`CodeDeployServiceRole`** die Berechtigungen zuzuweisen, die auf der von IAM verwalteten Richtlinie oder basieren. **`AWSCodeDeployRoleForLambda`** **`AWSCodeDeployRoleForLambdaLimited`** Beispiel:

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

   - Wenn Ihre Bereitstellung auf einer Amazon ECS-Rechenplattform erfolgt:

Rufen Sie den attach-role-policy Befehl auf, um der Servicerolle mit dem Namen **CodeDeployServiceRole** die Berechtigungen zuzuweisen, die auf der von IAM verwalteten Richtlinie **AWSCodeDeployRoleForECS** oder **AWSCodeDeployRoleForECSLimited** basieren. Beispiel:

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

Weitere Informationen zum Erstellen von Servicerollen finden Sie unter Erstellen einer Rolle für einen AWS Dienst im IAM-Benutzerhandbuch.

Rufen Sie die Servicerolle ARN ab (Konsole)

So verwenden Sie die IAM-Konsole, um den ARN der Servicerolle abzurufen:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Rollen.
3. Geben Sie in das Textfeld Filter **CodeDeployServiceRole** ein und drücken Sie dann die Eingabetaste.
4. Wählen Sie CodeDeployServiceRole.
5. Notieren Sie sich den Wert des Felds Role ARN.

Rufen Sie die Servicerolle ARN (CLI) ab

Um den ARN der AWS CLI Servicerolle abzurufen, rufen Sie den get-role Befehl für die Servicerolle mit dem Namen auf**CodeDeployServiceRole**:

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

Der zurückgegebene Wert ist der ARN der Servicerolle.

Schritt 3: Beschränken Sie die CodeDeploy Benutzerberechtigungen

Aus Sicherheitsgründen empfehlen wir, die Berechtigungen des Administratorbenutzers, den Sie erstellt haben, auf diejenigen Schritt 1: Einrichtung zu beschränken, die zum Erstellen und Verwalten von Bereitstellungen in CodeDeploy erforderlich sind.

Verwenden Sie die folgenden Verfahren, um die Berechtigungen des CodeDeploy Administratorbenutzers einzuschränken.

Bevor Sie beginnen

- Stellen Sie sicher, dass Sie in IAM Identity Center einen CodeDeploy Administratorbenutzer erstellt haben, indem Sie den Anweisungen unter folgen. Schritt 1: Einrichtung

So erstellen Sie einen Berechtigungssatz

Sie weisen diesen Berechtigungssatz später dem CodeDeploy Administratorbenutzer zu.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS IAM Identity Center Konsole unter https://console.aws.amazon.com/singlesignon/.
2. Wählen Sie im Navigationsbereich die Option Berechtigungssätze und dann Berechtigungssatz erstellen aus.
3. Wählen Sie Benutzerdefinierter Berechtigungssatz aus.
4. Wählen Sie Weiter aus.
5. Wählen Sie Inline-Richtlinie.
6. Entfernen Sie den Beispielcode.
7. Fügen Sie den folgenden Richtliniencode hinzu:

   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"
        }
    ]
}
```

Ersetzen Sie in dieser Richtlinie *arn:aws:iam::account-ID:role/CodeDeployServiceRole* durch den ARN-Wert der CodeDeploy Servicerolle, in der Sie

erstellt habenSchritt 2: Erstellen Sie eine Servicerolle für CodeDeploy. Sie finden den ARN-Wert auf der Detailseite der Servicerolle in der IAM-Konsole.

Mit der obigen Richtlinie können Sie eine Anwendung auf einer AWS Lambda-Rechenplattform, einer EC2/On-Premises Rechenplattform und einer Amazon ECS-Rechenplattform bereitstellen.

Sie können die in dieser Dokumentation bereitgestellten CloudFormation Vorlagen verwenden, um Amazon EC2 EC2-Instances zu starten, die kompatibel sind mit CodeDeploy. Um CloudFormation Vorlagen zum Erstellen von Anwendungen, Bereitstellungsgruppen oder Bereitstellungskonfigurationen zu verwenden, müssen Sie Zugriff auf CloudFormation— und AWS Dienste und Aktionen, die CloudFormation davon abhängen — gewähren, indem Sie die `cloudformation:*` entsprechende Berechtigung zur Berechtigungsrichtlinie des CodeDeploy Administratorbenutzers hinzufügen, etwa wie folgt:

JSON

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

8. Wählen Sie Weiter aus.
9. Geben Sie im Feld Name des Berechtigungssatzes Folgendes ein:

```
CodeDeployUserPermissionSet
```

10. Wählen Sie Weiter aus.
11. Überprüfen Sie auf der Seite Überprüfen und erstellen die Informationen und wählen Sie Erstellen aus.

Um den Berechtigungssatz dem CodeDeploy Administratorbenutzer zuzuweisen

1. Wählen Sie im Navigationsbereich das Kontrollkästchen neben dem aus AWS-Konten, bei dem Sie derzeit angemeldet sind AWS-Konto , und aktivieren Sie es anschließend.
2. Wählen Sie die Schaltfläche Benutzer oder Gruppen zuweisen.
3. Wählen Sie die Registerkarte Users.
4. Aktivieren Sie das Kontrollkästchen neben dem Benutzer mit CodeDeploy Administratorrechten.
5. Wählen Sie Weiter aus.
6. Aktivieren Sie das Kontrollkästchen neben`CodeDeployUserPermissionSet`.
7. Wählen Sie Weiter aus.
8. Überprüfen Sie die Informationen und wählen Sie Senden aus.

   Sie haben nun den CodeDeploy administrativen Benutzer `CodeDeployUserPermissionSet` zugewiesen und diese miteinander verknüpft. AWS-Konto

Um sich ab- und wieder als CodeDeploy Administratorbenutzer anzumelden

1. Bevor Sie sich abmelden, stellen Sie sicher, dass Sie die URL des AWS Zugriffsportals sowie den Benutzernamen und das Einmalkennwort für den CodeDeploy Administratorbenutzer haben.

   > **Note**
   >
   > Wenn Sie diese Informationen nicht haben, rufen Sie die Seite mit den CodeDeploy administrativen Benutzerdetails im IAM Identity Center auf und wählen Sie Passwort zurücksetzen, Einmalpasswort generieren [...] , und setzen Sie das Passwort erneut zurück, um die Informationen auf dem Bildschirm anzuzeigen.

2. Melden Sie sich ab AWS.
3. Fügen Sie die URL des AWS Zugangsportals in die Adressleiste Ihres Browsers ein.
4. Melden Sie sich als CodeDeploy administrativer Benutzer an.

   Auf dem AWS-KontoBildschirm erscheint ein Feld.
5. Wählen Sie AWS-Kontound wählen Sie dann den Namen des Benutzers, AWS-Konto dem Sie den CodeDeploy Administratorbenutzer und den Berechtigungssatz zugewiesen haben.
6. Wählen Sie neben dem die `CodeDeployUserPermissionSet` Option Verwaltungskonsole aus.

Das AWS-Managementkonsole erscheint. Sie sind jetzt als CodeDeploy administrativer Benutzer mit eingeschränkten Rechten angemeldet. Sie können jetzt als dieser Benutzer CodeDeploy-related Operationen und nur CodeDeploy-related Operationen ausführen.

Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances

> **Note**
>
> Wenn Sie die Amazon ECS- oder AWS Lambda-Rechenplattform verwenden, überspringen Sie diesen Schritt.

Ihre Amazon EC2 EC2-Instances benötigen eine Genehmigung für den Zugriff auf die Amazon S3 S3-Buckets oder GitHub Repositorys, in denen die Anwendungen gespeichert sind. Um Amazon EC2 EC2-Instances zu starten, die kompatibel sind mit CodeDeploy, müssen Sie eine zusätzliche IAM-Rolle, ein Instance-Profil, erstellen. Diese Anweisungen zeigen Ihnen, wie Sie ein IAM-Instance-Profil erstellen, das Sie an Ihre Amazon EC2 EC2-Instances anhängen können. Diese Rolle gibt dem CodeDeploy Agenten die Erlaubnis, auf die Amazon S3 S3-Buckets oder GitHub Repositorys zuzugreifen, in denen Ihre Anwendungen gespeichert sind.

Sie können ein IAM-Instance-Profil mit der AWS CLI, der IAM-Konsole oder den IAM-APIs erstellen.

> **Note**
>
> Sie können ein IAM-Instance-Profil einer Amazon-EC2-Instance beim Starten anfügen oder einer zuvor gestarteten Instance anfügen. Weitere Informationen finden Sie unter Instanzprofile.

Themen

- Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances (CLI)
- Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances (Konsole)

Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances (CLI)

Diese Anleitung setzt voraus, dass Sie bereits die Anweisungen in den ersten drei Schritten unter Erste Schritte mit CodeDeploy befolgt haben.

1. Erstellen Sie auf Ihrem Entwicklungscomputer eine Textdatei mit dem Namen `CodeDeployDemo-EC2-Trust.json`. Fügen Sie den folgenden Inhalt ein, damit Amazon EC2 in Ihrem Namen arbeiten kann:

   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. Erstellen Sie im selben Verzeichnis eine Textdatei mit dem Namen `CodeDeployDemo-EC2-Permissions.json`. Fügen Sie den folgenden Inhalt ein:

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

> **Note**
>
> Wir empfehlen, diese Richtlinie nur auf die Amazon S3 S3-Buckets zu beschränken, auf die Ihre Amazon EC2 EC2-Instances zugreifen müssen. Stellen Sie sicher, dass Sie Zugriff auf die Amazon S3 S3-Buckets gewähren, die den CodeDeploy Agenten enthalten. Andernfalls kann ein Fehler auftreten, wenn der CodeDeploy Agent auf den Instances installiert oder aktualisiert wird. Um dem IAM-Instance-Profil nur Zugriff auf einige CodeDeploy Resource Kit-Buckets in Amazon S3 zu gewähren, verwenden Sie die folgende Richtlinie, entfernen Sie jedoch die Zeilen für Buckets, auf die Sie den Zugriff verhindern möchten:
>
> ```
> {
>   "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

Wenn Sie die IAM-Autorisierung oder Amazon Virtual Private Cloud (VPC) -Endpoints mit verwenden möchten CodeDeploy, müssen Sie weitere Berechtigungen hinzufügen.

Weitere Informationen finden Sie unter Verwendung CodeDeploy mit Amazon Virtual Private Cloud.

3. Rufen Sie aus demselben Verzeichnis den create-role Befehl auf, um eine IAM-Rolle mit dem Namen zu erstellen**`CodeDeployDemo-EC2-Instance-Profile`**, der auf den Informationen in der ersten Datei basiert:

> **Important**
>
> Achten Sie darauf, dass `file://` vor dem Dateinamen steht. Dies ist bei diesem Befehl erforderlich.

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

4. Rufen Sie aus dem gleichen Verzeichnis den Befehl put-role-policy auf, um der Rolle namens **`CodeDeployDemo-EC2-Instance-Profile`** unter Verwendung der Informationen in der zweiten Datei die Berechtigungen zuzuweisen:

> **Important**
>
> Achten Sie darauf, dass `file://` vor dem Dateinamen steht. Dies ist bei diesem Befehl erforderlich.

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

5. Rufen Sie den attach-role-policy auf, um der Rolle Amazon EC2 Systems Manager Manager-Berechtigungen zu erteilen, damit SSM den CodeDeploy Agenten installieren kann. Diese Richtlinie ist nicht erforderlich, wenn Sie den Agenten über die Befehlszeile aus dem öffentlichen Amazon S3 S3-Bucket installieren möchten. Erfahren Sie mehr über die Installation des CodeDeploy -Agenten.

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

6. Rufen Sie den create-instance-profile Befehl gefolgt vom add-role-to-instance-profile Befehl auf, um ein IAM-Instance-Profil mit dem Namen **`CodeDeployDemo-EC2-Instance-Profile`** zu erstellen. Das Instance-Profil ermöglicht es Amazon EC2, die benannte IAM-Rolle an eine Amazon EC2 EC2-Instance **`CodeDeployDemo-EC2-Instance-Profile`** zu übergeben, wenn die Instance zum ersten Mal gestartet wird:

   ```
   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
   ```

   Wenn Sie den Namen des IAM-Instance-Profils benötigen, finden Sie weitere Informationen unter list-instance-profiles-for-role im Abschnitt IAM der Referenz.AWS CLI

Sie haben jetzt ein IAM-Instance-Profil erstellt, das Sie an Ihre Amazon EC2 EC2-Instances anhängen können. Weitere Informationen finden Sie unter IAM-Rollen für Amazon EC2 im Amazon-EC2-Benutzerhandbuch.

Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie in der IAM-Konsole im Navigationsbereich Richtlinien und dann Richtlinie erstellen aus.
3. Wählen Sie auf der Seite Berechtigungen angeben die Option JSON aus.
4. Entfernen Sie den `JSON` Beispielcode.
5. Fügen Sie folgenden Code ein:

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

```
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Wir empfehlen, diese Richtlinie nur auf die Amazon S3 S3-Buckets zu beschränken, auf die Ihre Amazon EC2 EC2-Instances zugreifen müssen. Stellen Sie sicher, dass Sie Zugriff auf die Amazon S3 S3-Buckets gewähren, die den CodeDeploy Agenten enthalten. Andernfalls kann ein Fehler auftreten, wenn der CodeDeploy Agent auf den Instances installiert oder aktualisiert wird. Um dem IAM-Instance-Profil nur Zugriff auf einige CodeDeploy Resource Kit-Buckets in Amazon S3 zu gewähren, verwenden Sie die folgende Richtlinie, entfernen Sie jedoch die Zeilen für Buckets, auf die Sie den Zugriff verhindern möchten:

```
{
  "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**
>
> Wenn Sie die IAM-Autorisierung oder Amazon Virtual Private Cloud (VPC) -Endpoints mit verwenden möchten CodeDeploy, müssen Sie weitere Berechtigungen hinzufügen. Weitere Informationen finden Sie unter Verwendung CodeDeploy mit Amazon Virtual Private Cloud.

6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Überprüfen und erstellen in das Feld Richtlinienname den Text ein**`CodeDeployDemo-EC2-Permissions`**.
8. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für die Richtlinie ein.
9. Wählen Sie Richtlinie erstellen aus.
10. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
11. Wählen Sie unter Anwendungsfall den EC2-Anwendungsfall aus.
12. Wählen Sie Weiter aus.

13. Aktivieren Sie in der Liste der Richtlinien das Kontrollkästchen neben der Richtlinie, die Sie gerade erstellt haben (CodeDeployDemo-EC2-Permissions). Verwenden Sie ggf. das Suchfeld, um die Richtlinie zu finden.
14. Um den CodeDeploy Agenten mithilfe von Systems Manager zu installieren oder zu konfigurieren, aktivieren Sie das Kontrollkästchen neben AmazonSSMManagedInstanceCore. Diese AWS verwaltete Richtlinie ermöglicht es einer Instanz, die Kernfunktionen des Systems Manager Manager-Service zu verwenden. Verwenden Sie ggf. das Suchfeld, um die Richtlinie zu finden. Diese Richtlinie ist nicht erforderlich, wenn Sie den Agenten über die Befehlszeile aus dem öffentlichen Amazon S3 S3-Bucket installieren möchten. Erfahren Sie mehr über die Installation des CodeDeploy-Agenten.
15. Wählen Sie Weiter aus.
16. Geben Sie auf der Seite Name, Überprüfung und Erstellung im Feld Rollenname einen Namen für die Servicerolle ein (z. B.**`CodeDeployDemo-EC2-Instance-Profile`**), und wählen Sie dann Rolle erstellen aus.

    Sie können unter Rollenbeschreibung auch eine Beschreibung für diese Servicerolle eingeben.

Sie haben jetzt ein IAM-Instance-Profil erstellt, das Sie an Ihre Amazon EC2 EC2-Instances anhängen können. Weitere Informationen finden Sie unter IAM-Rollen für Amazon EC2 im Amazon-EC2-Benutzerhandbuch.

Produkt- und Serviceintegrationen mit CodeDeploy

CodeDeploy Lässt sich standardmäßig in eine Reihe von AWS Diensten sowie Partnerprodukten und -diensten integrieren. Die folgenden Informationen können Ihnen bei der Konfiguration CodeDeploy für die Integration mit den von Ihnen verwendeten Produkten und Diensten helfen.

- Integration mit anderen AWS service
- Integration mit Produkten und Dienstleistungen von Partnern
- Integrationsbeispiele der Community

Integration mit anderen AWS service

CodeDeploy ist in die folgenden AWS Dienste integriert:

Amazon CloudWatch	Amazon CloudWatch ist ein Überwachungsdienst für AWS Cloud-Ressourcen und die Anwendungen, auf denen Sie laufen AWS. Sie können Amazon verwenden CloudWatch , um Metriken zu sammeln und zu verfolgen, Protokolldateien zu sammeln und zu überwachen und Alarme einzustellen. CodeDeploy unterstützt die folgenden CloudWatch Tools: • CloudWatch Alarme zur Überwachung Ihrer Bereitstellungen und zum Stoppen dieser Bereitstellungen, wenn Ihre angegebenen Überwachungsmetriken die in einer CloudWatch Alarmregel angegebenen Schwellenwerte über- oder unterschreiten. Um die Alarmüberwachung zu verwenden , richten Sie zunächst einen Alarm ein und fügen ihn dann der Anwendung oder Bereitstellungsgruppe CodeDeploy hinzu CloudWatch, in der Bereitstellungen beendet werden sollen, wenn der Alarm aktiviert wird.

Weitere Informationen:

- Erstellen von CloudWatch Protokollen und Alarmen

- Amazon CloudWatch Events zur Erkennung und Reaktion auf Änderungen im Status einer Instance oder einer Bereitstellung in Ihrem CodeDeploy Betrieb. Auf der Grundlage der von Ihnen erstellten Regeln ruft CloudWatch Events dann eine oder mehrere Zielaktionen auf, wenn eine Bereitstellung oder Instance in den Zustand übergeht, den Sie in einer Regel angeben.

  Weitere Informationen:

  - Überwachung von Bereitstellungen mit Amazon Events CloudWatch

- Amazon CloudWatch Logs zur Überwachung der drei Arten von Protokollen, die vom CodeDeploy Agenten erstellt wurden, ohne sich einzeln bei Instances anmelden zu müssen.

  Weitere Informationen:

  - CodeDeploy Logs in der CloudWatch Logs-Konsole anzeigen

Amazon EC2 Auto Scaling	CodeDeploy unterstützt Amazon EC2 Auto Scaling. Dieser AWS Service kann Amazon EC2 EC2-Instances anhand von Kriterien, die Sie angeben, automatisch starten, zum Beispiel: • Überschrittene Limits für die angegebene CPU-Auslastung. • Schreib- bzw. Lesevorgänge. • Ein- oder ausgehender Netzwerkverkehr in einem angegebenen Zeitintervall. Sie können eine Gruppe von Amazon EC2 EC2-Instances skalieren, wann immer Sie sie benötigen, und sie dann verwenden CodeDeploy , um automatisch Anwendungsrevisionen für sie bereitzustellen. Amazon EC2 Auto Scaling beendet diese Amazon EC2 EC2-Instances, wenn sie nicht mehr benötigt werden. Weitere Informationen: • Integration CodeDeploy mit Amazon EC2 Auto Scaling • Tutorial: Wird verwendet CodeDeploy , um eine Anwendung für eine Auto Scaling Scaling-Gruppe bereitzustellen • Unter der Haube: CodeDeploy und Auto Scaling Scaling-Integration

Amazon Elastic Container Service	Sie können CodeDeploy es verwenden, um eine containerisierte Amazon ECS-Anwendung als Task-Set bereitzustellen. CodeDeploy führt eine blue/green Bereitstellung durch, indem eine aktualisierte Version der Anwendung als neuer Ersatz-Tasksatz installiert wird. CodeDeploy leitet den Produktionsdatenverkehr vom ursprünglichen Anwendungs-Tasksatz zum Ersatz-Tasksatz um. Der ursprüngliche Tasksatz wird nach einer erfolgreichen Bereitstellung beendet. Weitere Informationen zu Amazon ECS finden Sie unter Amazon Elastic Container Service. Sie können die Art und Weise, wie der Datenverkehr während einer Bereitstellung auf den aktualisierten Aufgabensatz verschoben wird, verwalten, indem Sie eine Canary-, lineare oder All-at-once Konfiguration auswählen. Weitere Informationen zu Amazon ECS-Bereitstellungen finden Sie unter Bereitstellungen auf einer Amazon ECS-Rechenplattform.

AWS CloudTrail	CodeDeploy ist integriert in. AWS CloudTrail Dieser Service erfasst API-Aufrufe, die von oder im Namen Ihres AWS Kontos getätigt wurden, und übermittelt die Protokolldateien an einen von CodeDeploy Ihnen angegebenen Amazon S3 S3-Bucket. CloudTrailerfasst API-Aufrufe von der CodeDeploy Konsole, von CodeDeploy Befehlen über die AWS CLI oder direkt von den CodeDeploy APIs aus. Mithilfe der von CloudTrail erfassten Informationen können Sie Folgendes feststellen: • An welche Anfrage wurde gestellt CodeDeploy. • Die Quell-IP-Adresse, von der die Anfrage erfolgt ist. • Wer die Anfrage gestellt hat. • Wann sie gestellt wurde. Weitere Informationen: • Monitoring Deployments

AWS Cloud9	AWS Cloud9ist eine cloudbasierte integrierte Online-Entwicklungsumgebung (IDE), die Sie zum Schreiben, Ausführen, Debuggen und Bereitstellen von Code verwenden können, indem Sie nur einen Browser von einem mit dem Internet verbundenen Computer aus verwenden. AWS Cloud9 enthält einen Code-Editor, einen Debugger, ein Terminal und wichtige Tools wie The AWS CLI und Git. • Sie können die AWS Cloud9 IDE verwenden, um Code, der sich in einem GitHub Repository befindet, auszuführen, zu debuggen und zu erstellen. Sie können den Code im Fenster Environment (Umgebung) und den Editor-Registerkarten der IDE ansehen, ändern und speichern. Wenn Sie bereit sind, können Sie Git in der AWS Cloud9 Terminalsitzung verwenden, um Codeänderungen in Ihr GitHub Repository AWS CodeDeploy zu übertragen und dann Ihre Updates bereitzustellen. Weitere Informationen zur Verwendung von AWS Cloud9 with GitHub finden Sie unter GitHub Beispiel für AWS Cloud9. • Sie können die AWS Cloud9 IDE verwenden, um eine AWS Lambda Funktion zu aktualisieren. Sie können es dann verwenden AWS CodeDeploy , um eine Bereitstellung zu erstellen, die den Datenverkehr auf die neue Version Ihrer AWS Lambda Funktion umleitet. Weitere Informationen finden Sie unter Arbeiten mit AWS Lambda Funktionen in der AWS Cloud9 integrierten Entwicklungsumgebung (IDE).

	Weitere Informationen zu AWS Cloud9 finden Sie unter Was ist AWS Cloud9 und Erste Schritte mit AWS Cloud9.

AWS CodePipeline	AWS CodePipeline ist ein kontinuierlicher Bereitstellungsservice, mit dem Sie die für die Freigabe Ihrer Software in einem kontinuierlichen Bereitstellungsprozess erforderlichen Schritte entwickeln, visualisieren und automatisieren können. Sie können Ihren eigenen Veröffentlichungsprozess mit AWS CodePipeline definieren, damit der Service Ihren Code bei jeder Codeänderung erstellt, testet und bereitstellt. Es ist beispielsweise möglich, dass Sie drei Bereitstellungsgruppen für eine Anwendung haben: Beta, Gamma und Prod. Sie können eine Pipeline einrichten, sodass jedes Mal, wenn eine Änderung in Ihrem Quellcode vorgenommen wird, die Updates nacheinander für jede Bereitstellungsgruppe bereitgestellt werden. Sie können Folgendes AWS CodePipeline für die CodeDeploy Bereitstellung konfigurieren: • Code für Amazon EC2 EC2-Instances, lokale Instances oder beides. • Serverlose AWS Lambda-Funktionsversionen. Sie können die CodeDeploy Anwendung, Bereitstellung und Bereitstellungsgruppe, die in einer Bereitstellungsaktion verwendet werden sollen, entweder vor dem Erstellen der Pipeline oder im Assistenten „Pipeline erstellen“ erstellen. Weitere Informationen:

	• AWS Leitfaden für die ersten DevOps Schritte — Erfahren Sie, wie Sie CodePipeline mit CodeDeploy with kontinuierlich Quellcode in CodeCommit Repositorys für Amazon EC2 EC2-Instances bereitstellen und bereitstellen können. • Einfache Pipeline-Komplettlösung (Amazon S3 Bucket) • Einfache exemplarische Vorgehensweise für die Pipeline (Repository) CodeCommit • Four-stageAnleitung zur Pipeline
AWS Serverloses Anwendungsmodell	AWS Das Serverless Application Model (AWS SAM) ist ein Modell zur Definition serverloser Anwendungen. Es erweitert sich CloudFormation um eine vereinfachte Methode zur Definition von AWS Lambda Funktionen, Amazon API Gateway Gateway-APIs und Amazon DynamoDB-Tabellen, die für eine serverlose Anwendung erforderlich sind. Wenn Sie AWS SAM bereits verwenden, können Sie Bereitstellungseinstellungen hinzufügen, mit CodeDeploy denen Sie zunächst die Art und Weise verwalten können, wie der Datenverkehr während einer AWS Lambda-Anwendungsbereitstellung verlagert wird. Weitere Informationen finden Sie unter AWS Serverless Application Model.

Elastic Load Balancing	CodeDeploy unterstützt Elastic Load Balancing , einen Service, der den eingehenden Anwendungsdatenverkehr auf mehrere Amazon EC2 EC2-Instances verteilt. Bei CodeDeploy Bereitstellungen verhindern Load Balancer außerdem, dass Datenverkehr an Instances weitergeleitet wird, wenn diese nicht bereit sind, gerade in einer Umgebung bereitgestellt werden oder nicht mehr als Teil einer Umgebung benötigt werden. Weitere Informationen: • Integrating CodeDeploy with Elastic Load Balancing

Themen

- Integration CodeDeploy mit Amazon EC2 Auto Scaling
- Integration CodeDeploy mit Elastic Load Balancing

Integration CodeDeploy mit Amazon EC2 Auto Scaling

CodeDeploy unterstützt Amazon EC2 Auto Scaling, einen AWS Service, der Amazon EC2 EC2-Instances automatisch gemäß den von Ihnen definierten Bedingungen startet. Zu diesen Bedingungen können Grenzwerte gehören, die in einem bestimmten Zeitintervall für CPU-Auslastung, Lese- oder Schreibvorgänge auf Festplatten oder eingehenden oder ausgehenden Netzwerkverkehr überschritten wurden. Amazon EC2 Auto Scaling beendet die Instances, wenn sie nicht mehr benötigt werden. Weitere Informationen finden Sie unter Was ist Amazon EC2 Auto Scaling? im Benutzerhandbuch für Amazon EC2 Auto Scaling.

Wenn neue Amazon EC2-Instances als Teil einer Amazon EC2 Auto Scaling-Gruppe gestartet werden, CodeDeploy kann Ihre Revisionen automatisch für die neuen Instances bereitgestellt werden. Sie können Bereitstellungen auch CodeDeploy mit Amazon EC2 Auto Scaling Scaling-Instances koordinieren, die bei Elastic Load Balancing Load Balancers registriert sind. Weitere Informationen erhalten Sie unter Integrating CodeDeploy with Elastic Load Balancing und Richten Sie

in Elastic Load Balancing einen Load Balancer für CodeDeploy Amazon EC2 EC2-Bereitstellungen ein.

> **Note**
>
> Es können Probleme auftreten, wenn Sie mehrere Bereitstellungsgruppen einer einzelnen Amazon EC2 Auto Scaling Scaling-Gruppe zuordnen. Wenn eine Bereitstellung z. B. fehlschlägt, wird die Instance heruntergefahren. Bei den anderen Bereitstellungen, die ausgeführt wurden, kann jedoch eine Zeitüberschreitung von einer Stunde auftreten. Weitere Informationen finden Sie unter Vermeiden Sie es, mehrere Bereitstellungsgruppen einer einzigen Amazon EC2 Auto Scaling-Gruppe zuzuordnen und Under the hood: CodeDeploy and Amazon EC2 Auto Scaling integration.

Themen

- Bereitstellen von CodeDeploy Anwendungen für Amazon EC2 Auto Scaling Scaling-Gruppen
- Aktivierung von Terminierungsbereitstellungen bei Auto Scaling-Scale-In-Ereignissen
- So funktioniert Amazon EC2 Auto Scaling mit CodeDeploy
- Verwenden eines benutzerdefinierten AMI mit CodeDeploy Amazon EC2 Auto Scaling

Bereitstellen von CodeDeploy Anwendungen für Amazon EC2 Auto Scaling Scaling-Gruppen

So stellen Sie eine CodeDeploy Anwendungsrevision für eine Amazon EC2 Auto Scaling Scaling-Gruppe bereit:

1. Erstellen oder suchen Sie ein IAM-Instance-Profil, das es der Amazon EC2 Auto Scaling Scaling-Gruppe ermöglicht, mit Amazon S3 zu arbeiten. Weitere Informationen finden Sie unter Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances.

   > **Note**
   >
   > Sie können es auch verwenden CodeDeploy , um Revisionen aus GitHub Repositorys für Amazon EC2 Auto Scaling Scaling-Gruppen bereitzustellen. Amazon EC2 EC2-Instances benötigen zwar weiterhin ein IAM-Instance-Profil, für die Bereitstellung aus einem GitHub Repository sind jedoch keine zusätzlichen Berechtigungen erforderlich.

2. Erstellen oder verwenden Sie eine Amazon EC2 Auto Scaling Scaling-Gruppe und geben Sie das IAM-Instance-Profil in Ihrer Startkonfiguration oder Vorlage an. Weitere Informationen finden Sie unter IAM-Rolle für Anwendungen, die auf Amazon EC2 EC2-Instances ausgeführt werden.
3. Erstellen oder suchen Sie nach einer Servicerolle, die es ermöglicht, eine Bereitstellungsgruppe CodeDeploy zu erstellen, die die Amazon EC2 Auto Scaling Scaling-Gruppe enthält.
4. Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy und geben Sie den Namen der Amazon EC2 Auto Scaling Scaling-Gruppe, die Servicerolle und einige andere Optionen an. Für weitere Informationen siehe Erstellen Sie eine Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole) oder Erstellen Sie eine Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole).
5. Wird verwendet CodeDeploy , um Ihre Version in der Bereitstellungsgruppe bereitzustellen, die die Amazon EC2 Auto Scaling Scaling-Gruppe enthält.

Weitere Informationen finden Sie unter Tutorial: Wird verwendet CodeDeploy , um eine Anwendung für eine Auto Scaling Scaling-Gruppe bereitzustellen.

Aktivierung von Terminierungsbereitstellungen bei Auto Scaling-Scale-In-Ereignissen

Eine Terminierungsbereitstellung ist eine Art von CodeDeploy Bereitstellung, die automatisch aktiviert wird, wenn ein Auto Scaling-Scale-In-Ereignis eintritt. CodeDeploy führt die Terminierungsbereitstellung durch, unmittelbar bevor der Auto Scaling Scaling-Dienst die Instanz beendet. Stellt während einer Terminierungsbereitstellung CodeDeploy nichts bereit. Stattdessen generiert es Lebenszyklusereignisse, die Sie mit Ihren eigenen Skripten verknüpfen können, um benutzerdefinierte Shutdown-Funktionen zu aktivieren. Sie könnten das `ApplicationStop` Lebenszyklusereignis beispielsweise mit einem Skript verknüpfen, das Ihre Anwendung ordnungsgemäß herunterfährt, bevor die Instanz beendet wird.

Eine Liste der Lebenszyklusereignisse, die während einer Terminierungsbereitstellung CodeDeploy generiert werden, finden Sie unter. Verfügbarkeit von Hooks für Lebenszyklus-Ereignisse

Wenn die Bereitstellung zur Kündigung aus irgendeinem Grund fehlschlägt, kann die Instance-Kündigung fortgesetzt CodeDeploy werden. Das bedeutet, dass die Instance heruntergefahren wird, obwohl CodeDeploy nicht alle (oder einige) der Lifecycle-Ereignisse vollständig ausgeführt wurden.

Wenn Sie Terminierungsbereitstellungen nicht aktivieren, beendet der Auto Scaling-Service trotzdem Amazon EC2 EC2-Instances, wenn ein Scale-In-Ereignis eintritt, generiert aber keine CodeDeploy Lebenszyklusereignisse.

> **Note**
>
> Unabhängig davon, ob Sie Terminierungsbereitstellungen aktivieren oder nicht, kann es zu einem Wettlauf zwischen den durch Auto Scaling generierten Lebenszyklusereignissen und den Services kommen, wenn der Auto Scaling-Service eine Amazon EC2 EC2-Instance beendet, während eine CodeDeploy Bereitstellung läuft. CodeDeploy Beispielsweise kann das `Terminating` Lebenszyklusereignis (vom Auto Scaling Scaling-Dienst generiert) das `ApplicationStart` Ereignis (generiert durch die CodeDeploy Bereitstellung) überschreiben. In diesem Szenario kann es entweder bei der Kündigung der Amazon EC2 EC2-Instance oder bei der CodeDeploy Bereitstellung zu einem Fehler kommen.

Um Bereitstellungen CodeDeploy zur Ausführung von Terminierungen zu aktivieren

- Aktivieren Sie das Kontrollkästchen Einen Terminierungs-Hook zu Auto Scaling Scaling-Gruppen hinzufügen, wenn Sie Ihre Bereitstellungsgruppe erstellen oder aktualisieren. Anweisungen finden Sie Erstellen Sie eine Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole) unter oderErstellen Sie eine Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung (Konsole).

  Wenn Sie dieses Kontrollkästchen aktivieren CodeDeploy , wird ein Auto Scaling Scaling-Lifecycle-Hook in den Auto Scaling Scaling-Gruppen installiert, die Sie beim Erstellen oder Aktualisieren Ihrer CodeDeploy Bereitstellungsgruppe angeben. Dieser Hook wird Terminierungs-Hook genannt und ermöglicht Terminierungsbereitstellungen.

Nach der Installation des Terminations-Hooks läuft ein Scale-In-Ereignis (Terminierung) wie folgt ab:

1. Der Auto Scaling-Dienst (oder einfach Auto Scaling) stellt fest, dass ein Scale-In-Ereignis eintreten muss, und kontaktiert den EC2-Dienst, um eine EC2-Instance zu beenden.
2. Der EC2-Dienst beginnt mit dem Beenden der EC2-Instance. Die Instanz wechselt in den `Terminating` Status und dann in den Status. `Terminating:Wait`
3. `Terminating:Wait`Währenddessen führt Auto Scaling alle Lifecycle-Hooks aus, die an die Auto Scaling Scaling-Gruppe angehängt sind, einschließlich des Terminierungs-Hooks, der von installiert wurde CodeDeploy.
4. Der Termination-Hook sendet eine Benachrichtigung an die Amazon SQS SQS-Warteschlange, die von abgerufen wird. CodeDeploy

5. Nach Erhalt der Benachrichtigung CodeDeploy analysiert er die Nachricht, führt einige Validierungen durch und führt eine Terminierung durch.
6. Während die Terminierungsbereitstellung läuft, CodeDeploy sendet alle fünf Minuten Heartbeats an Auto Scaling, um es darüber zu informieren, dass an der Instance noch gearbeitet wird.
7. Bisher befindet sich die EC2-Instance immer noch im `Terminating:Wait` Status (oder möglicherweise im `Warmed:Pending:Wait` Status, wenn Sie Auto Scaling Scaling-Gruppen-Warm-Pools aktiviert haben).
8. Wenn die Bereitstellung abgeschlossen ist, CodeDeploy weist Auto Scaling auf `CONTINUE` den EC2-Kündigungsprozess hin, unabhängig davon, ob die Terminierungsbereitstellung erfolgreich war oder fehlgeschlagen ist.

So funktioniert Amazon EC2 Auto Scaling mit CodeDeploy

Wenn Sie eine CodeDeploy Bereitstellungsgruppe erstellen oder aktualisieren, sodass sie eine Auto Scaling Scaling-Gruppe enthält, CodeDeploy greift er über die CodeDeploy Service-Rolle auf die Auto Scaling Scaling-Gruppe zu und installiert dann Auto Scaling Scaling-Lebenszyklus-Hooks in Ihren Auto Scaling Scaling-Gruppen.

> **Note**
>
> Auto Scaling Scaling-Lifecycle-Hooks unterscheiden sich von den Lebenszyklusereignissen (auch als Lifecycle-Event-Hooks bezeichnet), die AppSpec Abschnitt „Hooks“ von diesem Handbuch generiert CodeDeploy und in diesem Handbuch beschrieben werden.

Folgende Auto Scaling Scaling-Lifecycle-Hooks werden CodeDeploy installiert:

- Ein Start-Hook — Dieser Hook benachrichtigt CodeDeploy , dass ein Auto Scale-Out-Ereignis im Gange ist und dass eine Startbereitstellung gestartet CodeDeploy werden muss.

  Während einer Startbereitstellung: CodeDeploy

  - Stellt eine Version Ihrer Anwendung auf der Scaled-Out-Instance bereit.
  - Generiert Lebenszyklusereignisse, um den Fortschritt der Bereitstellung anzuzeigen. Sie können diese Lebenszyklusereignisse mit Ihren eigenen Skripten verknüpfen, um benutzerdefinierte Startfunktionen zu aktivieren. Weitere Informationen finden Sie in der Tabelle unterVerfügbarkeit von Hooks für Lebenszyklus-Ereignisse.

Der Launch-Hook und die zugehörige Startbereitstellung sind immer aktiviert und können nicht ausgeschaltet werden.

- Ein Terminierungs-Hook — Dieser optionale Hook benachrichtigt CodeDeploy , dass ein Auto Scaling-Scale-In-Ereignis im Gange ist und dass eine Terminierungsbereitstellung gestartet CodeDeploy werden muss.

  CodeDeploy Generiert während einer Terminierungsbereitstellung Lebenszyklusereignisse, um den Fortschritt des Herunterfahrens der Instance anzuzeigen. Weitere Informationen finden Sie unter Aktivierung von Terminierungsbereitstellungen bei Auto Scaling-Scale-In-Ereignissen.

Themen

- Wie CodeDeploy werden die Lifecycle-Hooks nach der Installation verwendet?
- Wie CodeDeploy benennt Amazon EC2 Auto Scaling Scaling-Gruppen
- Reihenfolge der Ausführung von benutzerdefinierten Lifecycle-Hook-Ereignissen
- Scale-out Ereignisse während einer Bereitstellung
- Scale-in Ereignisse während einer Bereitstellung
- Reihenfolge der Ereignisse in AWS CloudFormation cfn-init-Skripte

Wie CodeDeploy werden die Lifecycle-Hooks nach der Installation verwendet?

Nachdem die Start- und Terminierungs-Lifecycle-Hooks installiert wurden, werden sie von jeweils CodeDeploy während Auto Scaling-Gruppen-Scale-Out- und Scale-In-Ereignissen verwendet.

Ein Scale-Out-Ereignis (Start) läuft wie folgt ab:

1. Der Auto Scaling-Dienst (oder einfach Auto Scaling) stellt fest, dass ein Scale-Out-Ereignis eintreten muss, und kontaktiert den EC2-Dienst, um eine neue EC2-Instance zu starten.
2. Der EC2-Dienst startet eine neue EC2-Instance. Die Instanz wechselt in den `Pending` Status und dann in den `Pending:Wait` Status.
3. `Pending:Wait`Währenddessen führt Auto Scaling alle Lifecycle-Hooks aus, die an die Auto Scaling Scaling-Gruppe angehängt sind, einschließlich des von installierten Start-Hooks CodeDeploy.
4. Der Launch-Hook sendet eine Benachrichtigung an die Amazon SQS SQS-Warteschlange, die von abgerufen wird. CodeDeploy

5. Nach Erhalt der Benachrichtigung CodeDeploy analysiert er die Nachricht, führt einige Validierungen durch und startet eine Startbereitstellung.
6. CodeDeploy Sendet bei laufender Startbereitstellung alle fünf Minuten Heartbeats an Auto Scaling, um das System darüber zu informieren, dass an der Instance noch gearbeitet wird.
7. Bisher befindet sich die EC2-Instance immer noch im `Pending:Wait` Status.
8. Wenn die Bereitstellung abgeschlossen ist, CodeDeploy zeigt Auto Scaling entweder `CONTINUE` `ABANDON` den EC2-Startvorgang an, je nachdem, ob die Bereitstellung erfolgreich war oder fehlgeschlagen ist.
   - Falls CodeDeploy angegeben`CONTINUE`, setzt Auto Scaling den Startvorgang fort und wartet entweder darauf, dass andere Hooks abgeschlossen sind, oder versetzt die Instance in den Status `Pending:Proceed` und dann in den `InService` Status.
   - Wenn CodeDeploy angegeben`ABANDON`, beendet Auto Scaling die EC2-Instance und startet den Startvorgang neu, falls erforderlich, um die gewünschte Anzahl von Instances zu erreichen, wie in der Einstellung Auto Scaling Desired Capacity definiert.

Ein Scale-In-Ereignis (Terminierung) läuft wie folgt ab:

Siehe Aktivierung von Terminierungsbereitstellungen bei Auto Scaling-Scale-In-Ereignissen.

Wie CodeDeploy benennt Amazon EC2 Auto Scaling Scaling-Gruppen

Bei blue/green Bereitstellungen auf einer EC2/On-Premises Rechenplattform haben Sie zwei Möglichkeiten, Instances zu Ihrer (grünen) Ersatzumgebung hinzuzufügen:

- Verwenden Sie Instances, die bereits vorhanden sind oder die Sie manuell erstellen.
- Verwenden Sie Einstellungen aus einer Amazon EC2 Auto Scaling Scaling-Gruppe, die Sie angeben, um Instances in einer neuen Amazon EC2 Auto Scaling Scaling-Gruppe zu definieren und zu erstellen.

Wenn Sie sich für die zweite Option entscheiden, wird eine CodeDeploy neue Amazon EC2 Auto Scaling Scaling-Gruppe für Sie bereitgestellt. Es verwendet die folgende Konventione für die Benennung der Gruppe:

```
CodeDeploy_deployment_group_name_deployment_id
```

Wenn beispielsweise eine Bereitstellung mit ID eine Bereitstellungsgruppe mit dem Namen `10` bereitstelltalpha-deployments, wird die bereitgestellte Amazon EC2 Auto Scaling Scaling-Gruppe benannt. `CodeDeploy_alpha-deployments_10` Weitere Informationen erhalten Sie unter Erstellen Sie eine Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung (Konsole) und GreenFleetProvisioningOption.

Reihenfolge der Ausführung von benutzerdefinierten Lifecycle-Hook-Ereignissen

Sie können Ihre eigenen Lifecycle-Hooks zu Amazon EC2 Auto Scaling Scaling-Gruppen hinzufügen, für die sie CodeDeploy bereitgestellt werden. Die Reihenfolge, in der diese benutzerdefinierten Lifecycle-Hook-Ereignisse ausgeführt werden, kann jedoch nicht im Verhältnis zu den CodeDeploy standardmäßigen Deployment-Lifecycle-Ereignissen vorgegeben werden. Wenn Sie beispielsweise einen benutzerdefinierten Lifecycle-Hook mit dem Namen `ReadyForSoftwareInstall` einer Amazon EC2 Auto Scaling Scaling-Gruppe hinzufügen, können Sie im Voraus nicht wissen, ob er vor dem ersten oder nach dem letzten CodeDeploy standardmäßigen Deployment-Lifecycle-Ereignis ausgeführt wird.

Informationen zum Hinzufügen von benutzerdefinierten Lifecycle-Hooks zu einer Amazon EC2 Auto Scaling Scaling-Gruppe finden Sie unter Hinzufügen von Lifecycle-Hooks im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Scale-out Ereignisse während einer Bereitstellung

Tritt während einer laufenden Bereitstellung ein Auto Scaling-Scale-Out-Ereignis auf, werden die neuen Instances mit der Anwendungsversion aktualisiert, die zuvor bereitgestellt wurde, und nicht mit der neuesten Anwendungsrevision. Wenn die Bereitstellung erfolgreich ist, werden die alten Instances und die neu skalierten Instances unterschiedliche Anwendungsrevisionen hosten. Um die Instanzen mit der älteren Version auf den neuesten Stand zu bringen, CodeDeploy wird automatisch eine Folgebereitstellung (unmittelbar nach der ersten) gestartet, um alle veralteten Instanzen zu aktualisieren. Wenn Sie dieses Standardverhalten ändern möchten, sodass veraltete EC2-Instances in der älteren Version verbleiben, finden Sie unter. Automatic updates to outdated instances

Wenn Sie die Scale-Out-Prozesse von Amazon EC2 Auto Scaling während Bereitstellungen aussetzen möchten, können Sie dies über eine Einstellung im `common_functions.sh` Skript tun, das für den Lastenausgleich mit verwendet wird. CodeDeploy Wenn `HANDLE_PROCS=true` die folgenden Auto Scaling Scaling-Ereignisse während des Bereitstellungsprozesses automatisch unterbrochen werden:

- AZRebalance

- AlarmNotification
- ScheduledActions
- ReplaceUnhealthy

> **Important**
>
> Nur die CodeDeployDefault.OneAtATime Bereitstellungskonfiguration unterstützt diese Funktionalität.

Weitere Informationen zur Verwendung `HANDLE_PROCS=true` zur Vermeidung von Bereitstellungsproblemen bei der Verwendung von Amazon EC2 Auto Scaling finden Sie unter Wichtiger Hinweis zur Handhabung von AutoScaling Prozessen in aws-codedeploy-samples auf. GitHub

Scale-in Ereignisse während einer Bereitstellung

Wenn eine Auto Scaling Scaling-Gruppe mit der Skalierung beginnt, während eine CodeDeploy Bereitstellung in dieser Auto Scaling Scaling-Gruppe im Gange ist, kann es zu einem Wettlauf zwischen dem Kündigungsprozess (einschließlich der Ereignisse im Lebenszyklus der Terminierung der Bereitstellung) und anderen CodeDeploy Lebenszyklusereignissen auf der CodeDeploy beendenden Instance kommen. Die Bereitstellung auf dieser speziellen Instance kann fehlschlagen, wenn die Instance beendet wird, bevor alle CodeDeploy Lebenszyklusereignisse abgeschlossen sind. Außerdem kann die gesamte CodeDeploy Bereitstellung fehlschlagen oder auch nicht, je nachdem, wie Sie in Ihrer Bereitstellungskonfiguration die Einstellung Mindestanzahl fehlerfreier Hosts festgelegt haben.

Reihenfolge der Ereignisse in AWS CloudFormation cfn-init-Skripte

Wenn Sie `cfn-init` (oder`cloud-init`) verwenden, um Skripts auf neu bereitgestellten Linux-based Instanzen auszuführen, schlagen Ihre Bereitstellungen möglicherweise fehl, es sei denn, Sie kontrollieren die Reihenfolge der Ereignisse, die nach dem Start der Instanz auftreten, genau.

Diese Reihenfolge muss folgendermaßen aussehen:

1. Die neu bereitgestellte Instance wird gestartet.
2. Alle `cfn-init`-Bootstrapping-Skripts werden abgeschlossen.
3. Der CodeDeploy Agent wird gestartet.

4. Die neueste Anwendungsrevision wird auf der Instance bereitgestellt.

Wenn die Reihenfolge der Ereignisse nicht sorgfältig kontrolliert wird, startet der CodeDeploy Agent möglicherweise eine Bereitstellung, bevor die Ausführung aller Skripts abgeschlossen ist.

Um die Reihenfolge der Ereignisse zu kontrollieren, verwenden Sie eine dieser bewährten Methoden:

- Installieren Sie den CodeDeploy Agenten über ein `cfn-init` Skript und platzieren Sie ihn hinter allen anderen Skripten.
- Binden Sie den CodeDeploy Agenten in ein benutzerdefiniertes AMI ein und verwenden Sie ein `cfn-init` Skript, um ihn zu starten. Platzieren Sie ihn hinter allen anderen Skripten.

Informationen zur Verwendung finden Sie *`cfn-init`* unter cfn-init im AWS CloudFormation Benutzerhandbuch.

Verwenden eines benutzerdefinierten AMI mit CodeDeploy Amazon EC2 Auto Scaling

Sie haben zwei Möglichkeiten, das Basis-AMI anzugeben, das verwendet werden soll, wenn neue Amazon EC2-Instances in einer Amazon EC2 Auto Scaling-Gruppe gestartet werden:

- Sie können ein benutzerdefiniertes Basis-AMI angeben, auf dem der CodeDeploy Agent bereits installiert ist. Da der Agent bereits installiert ist, startet diese Option neue Amazon EC2 EC2-Instances schneller als die andere Option. Diese Option erhöht jedoch die Wahrscheinlichkeit, dass erste Bereitstellungen von Amazon EC2 EC2-Instances fehlschlagen, insbesondere wenn der CodeDeploy Agent veraltet ist. Wenn Sie diese Option wählen, empfehlen wir Ihnen, den CodeDeploy Agenten in Ihrem benutzerdefinierten Basis-AMI regelmäßig zu aktualisieren.
- Sie können ein Basis-AMI angeben, bei dem der CodeDeploy Agent nicht installiert ist, und den Agenten installieren, wenn jede neue Instance in einer Amazon EC2 Auto Scaling Scaling-Gruppe gestartet wird. Diese Option startet neue Amazon EC2 EC2-Instances zwar langsamer als die andere Option, bietet jedoch eine höhere Wahrscheinlichkeit, dass die ersten Bereitstellungen von Instances erfolgreich sein werden. Bei dieser Methode wird die neueste Version des CodeDeploy-Agent verwendet.

Integration CodeDeploy mit Elastic Load Balancing

Während der CodeDeploy Bereitstellung verhindert ein Load Balancer, dass Internet-Traffic an Instances weitergeleitet wird, wenn diese nicht bereit sind, für die sie gerade bereitgestellt werden

oder als Teil einer Umgebung nicht mehr benötigt werden. Die genaue Rolle, die der Load Balancer spielt, hängt jedoch davon ab, ob er in einer blue/green Bereitstellung oder einer Bereitstellung vor Ort verwendet wird.

> **Note**
>
> Die Verwendung von Elastic Load Balancing Load Balancing-Load Balancern ist in blue/green Bereitstellungen obligatorisch und in In-Place-Bereitstellungen optional.

Arten von Elastic Load Balancing

Elastic Load Balancing bietet drei Arten von Load Balancern, die in CodeDeploy Bereitstellungen verwendet werden können: Classic Load Balancers, Application Load Balancers und Network Load Balancers.

Classic Load Balancer

Routen und Lastenausgleich entweder auf der Transportschicht (TCP/SSL) oder der Anwendungsebene (). HTTP/HTTPS Es unterstützt eine VPC.

> **Note**
>
> Classic Load Balancer werden bei Amazon ECS-Bereitstellungen nicht unterstützt.

Application Load Balancer

Leitet und verteilt die Last auf der Anwendungsebene (HTTP/HTTPS) und unterstützt pfadbasiertes Routing. Er kann Anforderungen an Ports auf jeder EC2-Instance oder Container-Instance in Ihrer Virtual Private Cloud (VPC) leiten.

> **Note**
>
> Die Application Load Balancer Balancer-Zielgruppen müssen den Zieltyp `instance` für Bereitstellungen auf EC2-Instances und `IP` für Fargate-Bereitstellungen haben. Weitere Informationen finden Sie unter Zieltyp.

Network Load Balancer

Routen und Lastenausgleich auf der Transportschicht (TCP/UDP Layer-4) basieren auf Adressinformationen, die aus dem TCP-Paket-Header und nicht aus dem Paketinhalt extrahiert wurden. Network Load Balancers können Datenverkehrsspitzen verarbeiten, die Quell-IP-Adresse des Clients beibehalten und eine feste IP für die Nutzungsdauer des Load Balancers verwenden.

Weitere Informationen zu Elastic Load Balancing Load Balancers finden Sie in den folgenden Themen:

- Was ist Elastic Load Balancing?
- Was ist ein Classic Load Balancer?
- Was ist ein Application Load Balancer?
- Was ist ein Network Load Balancer?

Blue/Green Bereitstellungen

Die Umleitung von Instance-Traffic hinter einem Elastic Load Balancing Load Balancer ist für Bereitstellungen von grundlegender Bedeutung. CodeDeploy blue/green

Während einer blue/green Bereitstellung ermöglicht der Load Balancer die Weiterleitung des Datenverkehrs an die neuen Instances in einer Bereitstellungsgruppe, in der die neueste Anwendungsversion bereitgestellt wurde (die Ersatzumgebung), gemäß den von Ihnen angegebenen Regeln, und blockiert dann den Datenverkehr von den alten Instances, auf denen die vorherige Anwendungsrevision lief (die ursprüngliche Umgebung).

Nachdem Instances in einer Ersatzumgebung bei einem oder mehreren Load Balancern registriert wurden, werden Instances aus der ursprünglichen Umgebung deregistriert und, falls Sie dies wünschen, beendet.

Für eine blue/green Bereitstellung können Sie eine oder mehrere Classic Load Balancer, Application Load Balancer-Zielgruppen oder Network Load Balancer Balancer-Zielgruppen in Ihrer Bereitstellungsgruppe angeben. Sie verwenden die CodeDeploy Konsole oder AWS CLI um die Load Balancer zu einer Bereitstellungsgruppe hinzuzufügen.

Weitere Informationen zu Load Balancern in blue/green Bereitstellungen finden Sie in den folgenden Themen:

- Richten Sie in Elastic Load Balancing einen Load Balancer für CodeDeploy Amazon EC2 EC2-Bereitstellungen ein
- Erstellen Sie eine Anwendung für eine blue/green Bereitstellung (Konsole)
- Erstellen Sie eine Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung (Konsole)

In-place Bereitstellungen

Während einer In-Situ-Bereitstellung verhindert ein Load Balancer, dass Internetdatenverkehr an eine Instance weitergeleitet wird, solange die Bereitstellung ausgeführt wird, und macht die Instance anschließend, wenn die Bereitstellung für sie abgeschlossen wurde, wieder für den Datenverkehr verfügbar.

Wenn in einer In-Situ-Bereitstellung kein Load Balancer verwendet wird, kann Internetdatenverkehr während des Bereitstellungsvorgangs an Instances geleitet werden. Ihre Kunden finden dann möglicherweise fehlerhafte, unvollständige oder veraltete Webanwendungen vor. Wenn Sie einen Elastic Load Balancing Load Balancer mit einer In-Place-Bereitstellung verwenden, werden Instances in einer Bereitstellungsgruppe vom Load Balancer abgemeldet, mit der neuesten Anwendungsversion aktualisiert und dann nach erfolgreicher Bereitstellung erneut beim Load Balancer als Teil derselben Bereitstellungsgruppe registriert. CodeDeploy wartet bis zu 1 Stunde, bis die Instance hinter dem Load Balancer wieder funktionsfähig ist. Wenn die Instance während der Wartezeit vom Load Balancer nicht als fehlerfrei markiert wird, wird CodeDeploy entweder zur nächsten Instance übergegangen oder die Bereitstellung schlägt aufgrund der Bereitstellungskonfiguration fehl.

Für eine direkte Bereitstellung können Sie eine oder mehrere Classic Load Balancer, Application Load Balancer-Zielgruppen oder Network Load Balancer Balancer-Zielgruppen angeben. Sie können die Load Balancer als Teil der Konfiguration der Bereitstellungsgruppe angeben, oder Sie können ein von bereitgestelltes Skript verwenden, um die Load Balancer CodeDeploy zu implementieren.

Geben Sie den Load Balancer für die direkte Bereitstellung mithilfe einer Bereitstellungsgruppe an

Um Load Balancer zu einer Bereitstellungsgruppe hinzuzufügen, verwenden Sie die CodeDeploy Konsole oder. AWS CLI Weitere Informationen zur Angabe eines Load Balancers in einer Bereitstellungsgruppe bei In-Situ-Bereitstellungen finden Sie in den folgenden Themen:

- Erstellen Sie eine Anwendung für eine direkte Bereitstellung (Konsole)
- Erstellen Sie eine Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole)

- Richten Sie in Elastic Load Balancing einen Load Balancer für CodeDeploy Amazon EC2 EC2-Bereitstellungen ein

Geben Sie den Load Balancer für die direkte Bereitstellung mithilfe eines Skripts an

Befolgen Sie die Schritte im folgenden Verfahren, um Bereitstellungslebenszyklusskripts zum Einrichten des Load Balancers für In-Situ-Bereitstellungen zu verwenden.

> **Note**
>
> Sie sollten die CodeDeployDefault.OneAtATime Bereitstellungskonfiguration nur verwenden, wenn Sie ein Skript verwenden, um einen Load Balancer für eine direkte Bereitstellung einzurichten. Gleichzeitige Ausführungen werden nicht unterstützt, und die CodeDeployDefault.OneAtATime Einstellung gewährleistet eine serielle Ausführung der Skripts. Weitere Informationen zu Bereitstellungskonfigurationen finden Sie unter Arbeiten mit Bereitstellungskonfigurationen in CodeDeploy.

Im CodeDeploy Samples-Repository unter finden Sie Anleitungen und Beispiele GitHub, die Sie an die Verwendung von CodeDeploy Elastic Load Balancing Balancing-Load Balancers anpassen können. Diese Repositorys enthalten drei Beispielskripte — `register_with_elb.shderegister_from_elb.sh`, und `common_functions.sh` —, die den gesamten Code enthalten, den Sie für den Einstieg benötigen. Bearbeiten Sie einfach die Platzhalter in diesen drei Skripts, und verweisen Sie dann aus der Datei `appspec.yml` auf diese Skripts.

Gehen Sie wie folgt vor, um In-Place-Bereitstellungen CodeDeploy mit Amazon EC2 EC2-Instances einzurichten, die bei Elastic Load Balancing Load Balancers registriert sind:

1. Laden Sie die Beispiele für den Load-Balancer-Typ herunter, den Sie für eine In-Situ-Bereitstellung verwenden möchten:
   - Classic Load Balancer
   - Application Load Balancer oder Network Load Balancer (dasselbe Skript kann für beide Typen verwendet werden)
2. Stellen Sie sicher, dass auf jeder Ihrer Amazon EC2 EC2-Zielinstanzen die AWS CLI installiert ist.

3. Stellen Sie sicher, dass jede Ihrer Amazon EC2 EC2-Ziel-Instances über ein IAM-Instance-Profil verfügt, das mindestens mit den Berechtigungen elasticloadbalancing: * und autoscaling: * verknüpft ist.
4. Fügen Sie im Quellcodeverzeichnis Ihrer Anwendung die Bereitstellungslebenszyklus-Ereignisskripts (`register_with_elb.sh`, `deregister_from_elb.sh` und `common_functions.sh`) hinzu.
5. Geben Sie in der Version `appspec.yml` für die Anwendung Anweisungen CodeDeploy zur Ausführung des Skripts während des ApplicationStartEreignisses und des `register_with_elb.sh` Skripts während des Ereignisses and`deregister_from_elb.sh`. ApplicationStop
6. Wenn die Instance Teil einer Amazon EC2 Auto Scaling Scaling-Gruppe ist, können Sie diesen Schritt überspringen.

   Im `common_functions.sh`-Skript:

   - Wenn Sie den Classic Load Balancer verwenden, geben Sie die Namen der Elastic Load Balancing Balancing-Load Balancer unter an `ELB_LIST=""` und nehmen Sie alle erforderlichen Änderungen an den anderen Bereitstellungseinstellungen in der Datei vor.
   - Wenn Sie den Application Load Balancer oder den Network Load Balancer verwenden, geben Sie die Namen der Elastic Load Balancing Balancing-Zielgruppen unter an und nehmen Sie alle erforderlichen Änderungen an den anderen Bereitstellungseinstellungen in der Datei vor. `TARGET_GROUP_LIST=""`
7. Bündeln Sie den Quellcode der Anwendung, die `appspec.yml` und die Bereitstellungslebenszyklus-Ereignisskripts in einer Anwendungsrevision, und laden Sie diese dann hoch. Stellen Sie die Revision auf den Amazon EC2 EC2-Instances bereit. Während der Bereitstellung melden die Skripte für das Deployment Lifecycle Event die Amazon EC2 EC2-Instance beim Load Balancer ab, warten, bis die Verbindung leer ist, und registrieren die Amazon EC2 EC2-Instance nach Abschluss der Bereitstellung erneut beim Load Balancer.

Integration mit Produkten und Dienstleistungen von Partnern

CodeDeploy verfügt über eine integrierte Integration für die folgenden Partnerprodukte und -dienstleistungen:

Ansible	Wenn Sie bereits über eine Reihe von Ansible-Playbooks verfügen, diese aber nur irgendwo

	ausführen können, CodeDeploy zeigt die Vorlage für Ansible, wie mit ein paar einfachen Bereitstellungs-Hooks sichergestellt werden kann, dass Ansible auf der lokalen Bereitste llungsinstanz verfügbar ist und die Playbooks ausführt. Wenn Sie bereits über einen Prozess zum Erstellen und Verwalten Ihres Inventars verfügen, gibt es auch ein Ansible-Modul, mit dem Sie den Agenten installieren und ausführen können. CodeDeploy Weitere Informationen: • Ansible und CodeDeploy

Atlassian — Bamboo und Bitbucket	Die CodeDeploy Aufgabe für Bamboo komprimiert das Verzeichnis, das eine AppSpec Datei enthält, in eine .zip-Datei, lädt die Datei auf Amazon S3 hoch und startet dann die Bereitstellung gemäß der in der Anwendung angegebenen Konfiguration. CodeDeploy CodeDeploy Mit der Bitbucket-Unterstützung von Atlassian kannst du bei Bedarf Code direkt von der Bitbucket-Benutzeroberfläche an jede deiner Bereitstellungsgruppen auf Amazon EC2 EC2-Instances übertragen. Das bedeutet, dass du dich nach der Aktualisierung des Codes in deinem Bitbucket-Repository nicht mehr bei deiner Continuous Integration (CI) -Plattform oder Amazon EC2 EC2-Instances anmelden musst, um einen manuellen Bereitstellungspro zess auszuführen. Weitere Informationen: • Die CodeDeploy Aufgabe für Bamboo verwenden • Ankündigung der Atlassian Bitbucket-Unterstützung für CodeDeploy

Chef	AWS bietet zwei Vorlagenbeispiele für die Integration von Chef und. CodeDeploy Das erste ist ein Chef-Kochbuch, das den CodeDeploy Agenten installiert und startet. Auf diese Weise können Sie Ihre Host-Infr astruktur mit Chef weiter verwalten, während Sie CodeDeploy verwenden. Die zweite Beispielvorlage zeigt, wie die Ausführung von Kochbüchern und Rezepten mit Chef-Solo auf jedem Knoten orchestriert wird. CodeDeploy Weitere Informationen: • Koch und CodeDeploy
CircleCI	CircleCI bietet ein Toolset für automatisierte Tests sowie fortlaufende Integration und Bereitstellung. Nachdem Sie eine IAM-Rolle für die Verwendung mit CircleCI erstellt und Ihre Bereitstellungsparameter in Ihrer circle.yml-Datei konfiguriert haben, können Sie CircleCI mit verwenden, um Anwendungsrevisionen CodeDeploy zu erstellen, sie in einen Amazon S3 S3-Bucket hochzuladen und dann Ihre Bereitstellungen zu initiieren und zu überwache n. AWS Weitere Informationen: • Verwenden einer CircleCI-Kugel zum Bereitstellen von Anwendungen für AWS CodeDeploy

CloudBees	Sie können das CodeDeploy Jenkins-Plugin, das auf CloudBeesDEV @cloud verfügbar ist, als Post-Build-Aktion verwenden. Sie können es beispielsweise am Ende einer fortlaufenden Bereitstellungs-Pipeline verwenden, um eine Anwendungsrevision für Ihre Server-Flotte bereitzustellen. Weitere Informationen: • CodeDeploy Das Jenkins-Plugin ist jetzt auf DEV @cloud verfügbar
Codeship	Sie können Codeship verwenden, um Anwendungsrevisionen bereitzustellen. CodeDeploy Mit der Codeship-Benutzero berfläche können Sie CodeDeploy einer Bereitstellungs-Pipeline für eine Niederlassung hinzufügen. Weitere Informationen: • Bereitstellen auf CodeDeploy • CodeDeploy Integration auf Codeship

GitHub	Sie können es verwenden CodeDeploy , um Anwendungsrevisionen aus Repositorys bereitzustellen. GitHub Sie können eine Bereitstellung auch von einem GitHub Repository aus auslösen, wenn der Quellcode in diesem Repository geändert wird. Weitere Informationen: • Integrieren CodeDeploy mit GitHub • Tutorial: Verwenden Sie CodeDeploy zum Bereitstellen einer Anwendung von GitHub • Automatisches Deployment GitHub über CodeDeploy
HashiCorp Konsul	Sie können das Open-Source-Tool HashiCorp Consul verwenden, um die Integrität und Stabilität Ihrer Anwendungsumgebung sicherzustellen, wenn Sie Anwendungen in bereitstellen. CodeDeploy Mit Consul können Sie Anwendungen registrieren, die während der Bereitstellung erkannt werden sollen, Anwendungen und Knoten in den Wartungsmodus setzen, um sie aus Bereitstellungen auszuschließen, und Bereitstellungen anhalten, wenn Ziel-Instances fehlerhaft werden. Weitere Informationen: • CodeDeploy Bereitstellungen mit Consul HashiCorp

Jenkins	Das CodeDeploy Jenkins-Plugin bietet einen Post-Build-Schritt für Ihr Jenkins-Projekt. Nach einem erfolgreichen Build wird der Workspace gezippt, auf Amazon S3 hochgeladen und eine neue Bereitstellung gestartet. Weitere Informationen: • CodeDeployJenkins-Plugin • Einrichtung des Jenkins-Plugins für CodeDeploy
Puppet Labs	AWS bietet Beispielvorlagen für Puppet und. CodeDeploy Das erste ist ein Puppet-Mo dul, das den CodeDeploy Agenten installiert und startet. Auf diese Weise können Sie Ihre Host-Infrastruktur mit Puppet weiter verwalten , während Sie CodeDeploy verwenden. Die zweite Beispielvorlage zeigt, wie die Ausführun g von Modulen und Manifesten mit einer Masterless-Mariponette auf jedem Knoten orchestriert wird. CodeDeploy Weitere Informationen: • Marionette und CodeDeploy

SaltStack	Sie können die SaltStackInfrastruktur mit CodeDeploy integrieren. Sie können das CodeDeploy Modul verwenden, um den CodeDeploy Agenten auf Ihren Minions zu installieren und auszuführen, oder Sie können es mit ein paar einfachen Bereitstellungs-Hooks verwenden, CodeDeploy um den Betrieb Ihrer Salt States zu orchestrieren. Weitere Informationen: • SaltStack und CodeDeploy
TeamCity	Sie können das CodeDeploy Runner-Plugin verwenden, um Anwendungen direkt von dort aus bereitzustellen. TeamCity Das Plugin fügt einen TeamCity Build-Schritt hinzu, der eine Anwendungsrevision vorbereitet und in einen Amazon S3 S3-Bucket hochlädt, die Revision in einer CodeDeploy Anwendung registriert, eine CodeDeploy Bereitstellung erstellt und, falls Sie dies wünschen, wartet, bis die Bereitstellung abgeschlossen ist. Weitere Informationen: • CodeDeploy Runner (Herunterladen) • CodeDeploy Runner-Plugin (Dokumentation)
Travis CI	Sie können Travis CI so konfigurieren, dass CodeDeploy nach einem erfolgreichen Build eine Bereitstellung ausgelöst wird. Weitere Informationen: • Travis CI und Bereitstellungen CodeDeploy

Themen

- Integrieren CodeDeploy mit GitHub

Integrieren CodeDeploy mit GitHub

CodeDeploy unterstützt GitHub, einen webbasierten Dienst zum Hosten und Teilen von Code. CodeDeploy kann Anwendungsrevisionen, die in GitHub Repositorys oder Amazon S3 S3-Buckets gespeichert sind, für Instances bereitstellen. CodeDeploy unterstützt nur GitHub Bereitstellungen EC2/On-Premises .

Themen

- CodeDeploy Revisionen werden bereitgestellt von GitHub
- GitHub Verhalten mit CodeDeploy

CodeDeploy Revisionen werden bereitgestellt von GitHub

So stellen Sie eine Anwendungsrevision aus einem GitHub Repository auf Instanzen bereit:

1. Erstellen Sie eine Version, die kompatibel ist mit CodeDeploy und dem Amazon EC2 EC2-Instance-Typ, für den Sie die Bereitstellung durchführen möchten.

   Zum Erstellen einer kompatiblen Revision befolgen Sie die Anweisungen in Planen Sie eine Überarbeitung für CodeDeploy und Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy.
2. Verwenden Sie ein GitHub Konto, um Ihre Version zu einem GitHub Repository hinzuzufügen.

   Informationen zum Erstellen eines GitHub Kontos finden Sie unter Beitreten GitHub. Informationen zum Erstellen eines GitHub Repositorys finden Sie unter Repo erstellen.
3. Verwenden Sie die Seite „Bereitstellung erstellen“ in der CodeDeploy Konsole oder den AWS CLI create-deployment Befehl, um Ihre Revision aus Ihrem GitHub Repository auf Ziel-Instances bereitzustellen, die für die Verwendung in CodeDeploy Bereitstellungen konfiguriert sind.

   Wenn Sie den create-deployment Befehl aufrufen möchten, müssen Sie zunächst GitHub auf der Seite „Bereitstellung erstellen“ der Konsole die CodeDeploy Erlaubnis erteilen, im Namen Ihres bevorzugten GitHub Kontos mit der angegebenen Anwendung zu interagieren. Sie müssen dies nur einmal für jede Anwendung tun.

Informationen zur Verwendung der Seite „Bereitstellung erstellen“ für die Bereitstellung aus einem GitHub Repository finden Sie unterErstellen Sie eine Bereitstellung mit CodeDeploy.

Informationen zum Aufrufen des create-deployment Befehls zur Bereitstellung aus einem GitHub Repository finden Sie unterErstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (CLI).

Informationen zum Vorbereiten von Instanzen für die Verwendung in CodeDeploy Bereitstellungen finden Sie unterArbeiten mit Instanzen für CodeDeploy.

Weitere Informationen finden Sie unter Tutorial: Verwenden Sie CodeDeploy zum Bereitstellen einer Anwendung von GitHub.

GitHub Verhalten mit CodeDeploy

Themen

- GitHub Authentifizierung mit Anwendungen in CodeDeploy
- CodeDeploy Interaktion mit privaten und öffentlichen Repositorien GitHub
- CodeDeploy Interaktion mit von der Organisation verwalteten Repositorys GitHub
- Automatisch bereitstellen von CodePipeline mit CodeDeploy

GitHub Authentifizierung mit Anwendungen in CodeDeploy

Nachdem Sie die CodeDeploy Erlaubnis zur Interaktion erteilt haben GitHub, wird die Verknüpfung zwischen diesem GitHub Konto und der Anwendung in gespeichert CodeDeploy. Sie können die Anwendung mit einem anderen GitHub Konto verknüpfen. Sie können auch die Erlaubnis CodeDeploy zur Interaktion mit widerrufen GitHub.

Um ein GitHub Konto mit einer Anwendung zu verknüpfen in CodeDeploy

1. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie die Anwendung aus, die Sie mit einem anderen GitHub Konto verknüpfen möchten.
4. Wenn Ihre Anwendung keine Bereitstellungsgruppe hat, wählen Sie Bereitstellungsgruppe erstellen, um eine zu erstellen. Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy. Eine Bereitstellungsgruppe ist erforderlich, um im nächsten Schritt Bereitstellung erstellen auszuwählen.
5. Wählen Sie unter Bereitstellungen die Option Bereitstellung erstellen.

> **Note**
>
> Sie müssen keine neue Bereitstellung erstellen. Dies ist derzeit die einzige Möglichkeit, ein anderes GitHub Konto mit einer Anwendung zu verknüpfen.

6. Wählen Sie in den Bereitstellungseinstellungen für Revisionstyp die Option Meine Anwendung ist gespeichert in aus GitHub.
7. Führen Sie eine der folgenden Aktionen aus:

   - Um eine Verbindung für AWS CodeDeploy Anwendungen mit einem GitHub Konto herzustellen, melden Sie sich auf einer separaten Registerkarte GitHub im Webbrowser ab. Geben Sie im Feld GitHub Tokenname einen Namen zur Identifizierung dieser Verbindung ein, und wählen Sie dann Connect aus GitHub. Auf der Webseite werden Sie aufgefordert, die Interaktion mit GitHub Ihrer CodeDeploy Anwendung zu autorisieren. Fahren Sie fort mit Schritt 10.
   - Um eine Verbindung zu verwenden, die Sie bereits erstellt haben, wählen Sie im Feld GitHubTokenname den Namen aus, und klicken Sie dann auf Connect GitHub. Fahren Sie fort mit Schritt 8.
   - Um eine Verbindung zu einem anderen GitHub Konto herzustellen, melden Sie sich GitHub in einem separaten Webbrowser-Tab ab. Geben Sie im Feld GitHub Tokenname einen Namen zur Identifizierung der Verbindung ein, und wählen Sie dann Connect aus GitHub. Auf der

Webseite werden Sie aufgefordert, die Interaktion mit GitHub Ihrer CodeDeploy Anwendung zu autorisieren. Fahren Sie fort mit Schritt 10.

8. Wenn Sie noch nicht angemeldet sind GitHub, folgen Sie den Anweisungen auf der Anmeldeseite, um sich mit dem GitHub Konto anzumelden, mit dem Sie die Anwendung verknüpfen möchten.
9. Wählen Sie Anwendung autorisieren aus. GitHub CodeDeployerteilt die Erlaubnis, im Namen des GitHub angemeldeten GitHub Kontos mit der ausgewählten Anwendung zu interagieren.
10. Wenn Sie keine Bereitstellung erstellen möchten, wählen Sie Cancel.

Um die Erlaubnis zur Interaktion zu CodeDeploy widerrufen GitHub

1. Melden Sie sich GitHub mit den Anmeldeinformationen für das GitHub Konto an, für das Sie die AWS CodeDeploy Erlaubnis widerrufen möchten.
2. Öffnen Sie die Seite GitHub Anwendungen, suchen Sie CodeDeployin der Liste der autorisierten Anwendungen nach, und folgen Sie dann den Anweisungen GitHub zum Widerrufen der Autorisierung für eine Anwendung.

CodeDeploy Interaktion mit privaten und öffentlichen Repositorien GitHub

CodeDeploy unterstützt die Bereitstellung von Anwendungen aus privaten und öffentlichen GitHub Repositorien. Wenn Sie in Ihrem Namen die CodeDeploy Erlaubnis zum Zugriff GitHub erteilen, CodeDeploy hat er Lese- und Schreibzugriff auf alle privaten GitHub Repositorys, auf die Ihr GitHub Konto Zugriff hat. Liest jedoch CodeDeploy nur aus Repositorys. GitHub Es wird in keines Ihrer privaten GitHub Repositorys schreiben.

CodeDeploy Interaktion mit von der Organisation verwalteten Repositorys GitHub

Standardmäßig gewähren GitHub Repositorys, die von einer Organisation verwaltet werden (im Gegensatz zu den privaten oder öffentlichen Repositorys Ihres Kontos), keinen Zugriff auf Anwendungen von Drittanbietern, einschließlich. CodeDeploy Ihre Bereitstellung schlägt fehl, wenn die Einschränkungen für Drittanbieteranwendungen einer Organisation aktiviert sind GitHub und Sie versuchen, Code aus ihrem GitHub Repository bereitzustellen. Dieses Problem lässt sich auf zweierlei Weise lösen.

- Als Mitglied der Organisation können Sie den Eigentümer der Organisation bitten, den Zugriff auf CodeDeploy zu genehmigen. Die Schritte zur Beantragung dieses Zugriffs hängen davon ab, ob Sie bereits eine Autorisierung CodeDeploy für Ihr individuelles Konto vorgenommen haben:

  - Wenn Sie CodeDeploy in Ihrem Konto autorisierten Zugriff darauf haben, finden Sie weitere Informationen unter Beantragung der Genehmigung Ihrer autorisierten Anwendungen durch die Organisation.
  - Wenn Sie CodeDeploy in Ihrem Konto noch keinen autorisierten Zugriff darauf haben, finden Sie weitere Informationen unter Beantragung der Genehmigung durch die Organisation für Anwendungen von Drittanbietern.
- Der Eigentümer der Organisation kann alle für Drittanbieteranwendungen geltenden Beschränkungen der Organisation deaktivieren. Weitere Informationen finden Sie unter Einschränkungen für Drittanbieteranwendungen für Ihr Unternehmen deaktivieren.

Weitere Informationen finden Sie unter Einschränkungen für Drittanbieteranwendungen.

Automatisch bereitstellen von CodePipeline mit CodeDeploy

Sie können eine Bereitstellung von a aus auslösen CodePipeline , wenn sich der Quellcode ändert. Weitere Informationen finden Sie unter CodePipeline.

Integrationsbeispiele der Community

In den folgenden Abschnitten werden Links zu Blog-Posts, Artikel und von der Community bereitgestellte Beispiele vorgestellt.

> **Note**
>
> Diese Links dienen nur zu Informationszwecken und sollten nicht als umfassende Liste oder als Bestätigung für den Inhalt der Beispiele gelten. AWS ist nicht für den Inhalt oder die Genauigkeit externer Inhalte verantwortlich.

Blog-Posts

- Automatisieren Sie die Bereitstellung in CodeDeploy CloudFormation

  Erfahren Sie, wie Sie die Bereitstellung einer Anwendung bereitstellen, CodeDeploy indem Sie CloudFormation

  Veröffentlicht im Januar 2016

- AWS Toolkit for Eclipse Integration mit CodeDeploy (Teil 1)

  AWS Toolkit for Eclipse Integration mit CodeDeploy (Teil 2)

  AWS Toolkit for Eclipse Integration mit CodeDeploy (Teil 3)

  Erfahren Sie, wie Java-Entwickler das CodeDeploy Plugin für Eclipse verwenden können, um Webanwendungen AWS direkt aus Eclipse-Entwicklungsumgebungen bereitzustellen.

  Veröffentlichung Februar 2015
- Automatisches Deployment GitHub über CodeDeploy

  Erfahren Sie, wie automatische Bereitstellungen von GitHub bis verwendet werden CodeDeploy können, um eine durchgängige Pipeline zu erstellen — von der Quellcodeverwaltung bis hin zu Ihren Test- oder Produktionsumgebungen.

  Erschienen: Dezember 2014

CodeDeploy Anleitungen

In diesem Abschnitt finden Sie verschiedene Tutorials, anhand derer Sie den Umgang mit CodeDeploy erlernen können.

Die Verfahren in diesen Tutorials enthalten Vorschläge für den Speicherort von Dateien (z. B. c:\temp) und die Namen, die Buckets, Unterordnern oder Dateien gegeben werden sollen (z. B. amzn-s3-demo-bucket HelloWorldApp CodeDeployDemo-EC2-Trust.json, bzw.). Sie müssen sie jedoch nicht verwenden. Sie können stattdessen an den entsprechenden Stellen eigene Dateispeicherorte und Namen angeben.

Themen

- Tutorial: Bereitstellung WordPress auf einer Amazon EC2 EC2-Instance (Amazon Linux oder Red Hat Enterprise Linux und Linux, macOS oder Unix)
- Tutorial: Stellen Sie ein „Hallo, Welt!“ bereit Anwendung mit CodeDeploy (Windows Server)
- Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat Enterprise Linux)
- Tutorial: Wird verwendet CodeDeploy , um eine Anwendung für eine Auto Scaling Scaling-Gruppe bereitzustellen
- Tutorial: Verwenden Sie CodeDeploy zum Bereitstellen einer Anwendung von GitHub
- Tutorial: Bereitstellen einer Anwendung in Amazon ECS
- Tutorial: Bereitstellen eines Amazon ECS-Service mit einem Validierungstest
- Tutorial: Stellen Sie eine aktualisierte Lambda-Funktion bereit mit CodeDeploy und AWS Serverloses Anwendungsmodell

Tutorial: Bereitstellung WordPress auf einer Amazon EC2 EC2-Instance (Amazon Linux oder Red Hat Enterprise Linux und Linux, macOS oder Unix)

In diesem Tutorial stellen WordPress Sie ein Open-Source-Blogging-Tool und ein Content-Management-System auf der Basis von PHP und MySQL auf einer einzelnen Amazon EC2 EC2-Instance bereit, auf der Amazon Linux oder Red Hat Enterprise Linux (RHEL) ausgeführt wird.

Nicht das, wonach Sie gesucht haben?

- Informationen zum Bereitstellen auf einer Amazon EC2 EC2-Instance, auf der stattdessen Windows Server ausgeführt wird, finden Sie unterTutorial: Stellen Sie ein „Hallo, Welt!“ bereit Anwendung mit CodeDeploy (Windows Server).
- Informationen zur Bereitstellung auf einer lokalen Instance statt auf einer Amazon EC2 EC2-Instance finden Sie unter. Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat Enterprise Linux)

Die Schritte dieses Tutorials werden aus der Perspektive einer lokalen Entwicklungsmaschine vorgestellt, auf der Linux, macOS oder Unix ausgeführt wird. Sie können die meisten der Schritte auch auf einem lokalen Computer mit Windows durchführen, müssen dann jedoch alle Schritte anpassen, die Befehle wie chmod und wget, Anwendungen wie sed und Verzeichnispfade wie `/tmp` verwenden.

Bevor Sie mit diesem Tutorial beginnen, müssen Sie sicherstellen, dass die unter Erste Schritte mit CodeDeploy beschriebenen Voraussetzungen erfüllt sind. Dazu gehören die Konfiguration eines Benutzers, die Installation oder Aktualisierung von und die AWS CLI Erstellung eines IAM-Instanzprofils und einer Servicerolle.

Themen

- Schritt 1: Starten und konfigurieren Sie eine Amazon Linux- oder Red Hat Enterprise Linux Amazon EC2 EC2-Instance
- Schritt 2: Konfigurieren Sie Ihren Quellinhalt für die Bereitstellung auf der Amazon Linux- oder Red Hat Enterprise Linux Amazon EC2 EC2-Instance
- Schritt 3: Laden Sie Ihre WordPress Bewerbung auf Amazon S3 hoch
- Schritt 4: Stellen Sie Ihre WordPress Anwendung bereit
- Schritt 5: Aktualisieren Sie Ihre WordPress Anwendung und stellen Sie sie erneut bereit
- Schritt 6: Bereinigen Sie Ihre WordPress Anwendung und die zugehörigen Ressourcen

Schritt 1: Starten und konfigurieren Sie eine Amazon Linux- oder Red Hat Enterprise Linux Amazon EC2 EC2-Instance

Um die WordPress Anwendung bereitzustellen CodeDeploy, benötigen Sie eine Amazon EC2 EC2-Instance, auf der Amazon Linux oder Red Hat Enterprise Linux (RHEL) ausgeführt wird. Die Amazon EC2 EC2-Instance erfordert eine neue Sicherheitsregel für eingehenden Datenverkehr, die HTTP-

Verbindungen zulässt. Diese Regel ist erforderlich, um die WordPress Seite nach erfolgreicher Bereitstellung in einem Browser anzuzeigen.

Folgen Sie den Anweisungen in Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy. Wenn Sie zu dem Teil in diesen Anweisungen gelangen, in dem es darum geht, der Instance ein Amazon EC2 EC2-Instance-Tag zuzuweisen, geben Sie unbedingt den Tag-Schlüssel von **`Name`** und den Tag-Wert von an. **`CodeDeployDemo`** (Wenn Sie einen anderen Tag-Schlüssel oder Tag-Wert angeben, dann können die Anweisungen in Schritt 4: Stellen Sie Ihre WordPress Anwendung bereit zu unerwarteten Ergebnissen führen.)

Nachdem Sie die Anweisungen zum Starten der Amazon EC2 EC2-Instance befolgt haben, kehren Sie zu dieser Seite zurück und fahren Sie mit dem nächsten Abschnitt fort. Fahren Sie nicht mit Erstellen Sie eine Anwendung mit CodeDeploy dem nächsten Schritt fort.

Connect zu Ihrer Amazon Linux- oder RHEL Amazon EC2 EC2-Instance her

Nachdem Ihre neue Amazon EC2 EC2-Instance gestartet wurde, folgen Sie diesen Anweisungen, um zu üben, wie Sie sich mit ihr verbinden.

1. Verwenden Sie den ssh Befehl (oder einen SSH-capable Terminalemulator wie PuTTY), um eine Verbindung zu Ihrer Amazon Linux- oder RHEL Amazon EC2 EC2-Instance herzustellen. Sie benötigen die öffentliche DNS-Adresse der Instance und den privaten Schlüssel für das key pair, das Sie beim Starten der Amazon EC2 EC2-Instance verwendet haben. Weitere Informationen finden Sie unter Connect to Your Instance (Verbindung mit der Instance herstellen).

   Wenn die öffentliche DNS-Adresse beispielsweise lautet **`ec2-01-234-567-890.compute-1.amazonaws.com`** und Ihr Amazon EC2 EC2-Instance-Schlüsselpaar für den SSH-Zugriff benannt ist**`codedeploydemo.pem`**, würden Sie Folgendes eingeben:

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

   /path/to/`codedeploydemo.pem`Ersetzen Sie durch den Pfad zu Ihrer `.pem` Datei und die Beispiel-DNS-Adresse durch die Adresse Ihrer Amazon Linux- oder RHEL Amazon EC2 EC2-Instance.

> **Note**
>
> Wenn ein Fehler angezeigt wird, der besagt, dass die Berechtigungen der Schlüsseldatei nicht restriktiv genug sind, müssen Sie diese Berechtigungen einschränken, sodass nur der aktuelle Benutzer (Sie) Zugriff hat. Geben Sie beispielsweise mit dem chmod Befehl unter Linux, macOS oder Unix Folgendes ein:

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

2. Nachdem Sie sich angemeldet haben, sehen Sie das AMI-Banner für die Amazon EC2 EC2-Instance. Für Amazon Linux sollte es so aussehen:

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

3. Sie können sich jetzt von der laufenden Amazon EC2 EC2-Instance abmelden.

> **Warning**
>
> Beenden oder beenden Sie die Amazon EC2 EC2-Instance nicht. Andernfalls kann sie CodeDeploy nicht bereitgestellt werden.

Fügen Sie eine Regel für eingehenden Datenverkehr hinzu, die HTTP-Verkehr zu Ihrer Amazon Linux- oder RHEL Amazon EC2 EC2-Instance zulässt

Im nächsten Schritt wird bestätigt, dass Ihre Amazon EC2 EC2-Instance über einen offenen HTTP-Port verfügt, sodass Sie die Startseite der bereitgestellten WordPress Anwendung in einem Browser sehen können.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie Instances und dann Ihre Instance aus.

3. Wählen Sie auf der Registerkarte Beschreibung unter Sicherheitsgruppen die Option Regeln für eingehenden Datenverkehr anzeigen aus.

   In Ihrer Sicherheitsgruppe sollte eine Liste mit Regeln wie die folgende angezeigt werden:

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

4. Wählen Sie unter Sicherheitsgruppen die Sicherheitsgruppe für Ihre Amazon EC2 EC2-Instance aus. Ein möglicher Name ist **`launch-wizard-N`**. Beim ***N*** im Namen handelt es sich um eine Nummer, die Ihrer Sicherheitsgruppe beim Erstellen Ihrer Instance zugewiesen wurde.

   Wählen Sie die Registerkarte Inbound (Eingehend). Wenn die Sicherheitsgruppe für Ihre Instance korrekt konfiguriert ist, sollte Ihnen eine Regel mit den folgenden Werten angezeigt werden:

   - Typ: HTTP
   - Protocol (Protokoll): TCP
   - Portbereich: 80
   - Quelle: 0.0.0. 0/0

5. Wenn Sie keine Regel mit diesen Werten sehen, verwenden Sie die Verfahren unter Regeln zu einer Sicherheitsgruppe hinzufügen, um sie einer neuen Sicherheitsregel hinzuzufügen.

Schritt 2: Konfigurieren Sie Ihren Quellinhalt für die Bereitstellung auf der Amazon Linux- oder Red Hat Enterprise Linux Amazon EC2 EC2-Instance

Jetzt ist es an der Zeit, die Quellinhalte Ihrer Anwendung zu konfigurieren, damit Sie Inhalte haben, die Sie auf der Instance bereitstellen können.

Themen

- Holen Sie sich den Quellcode
- Erstellen Sie Skripts, um Ihre Anwendung auszuführen
- Fügen Sie eine Anwendungsspezifikationsdatei hinzu

Holen Sie sich den Quellcode

In diesem Tutorial stellen Sie die Plattform zur Veröffentlichung von WordPress Inhalten von Ihrem Entwicklungscomputer auf der Amazon EC2 EC2-Zielinstanz bereit. Um den WordPress Quellcode abzurufen, können Sie integrierte Befehlszeilenaufrufe verwenden. Wenn Sie auf Ihrem Entwicklungscomputer Git installiert haben, können Sie stattdessen auch Git verwenden.

Für diese Schritte gehen wir davon aus, dass Sie eine Kopie des WordPress Quellcodes in das `/tmp` Verzeichnis auf Ihrem Entwicklungscomputer heruntergeladen haben. (Sie können jedes beliebige Verzeichnis wählen. Denken Sie aber daran, den Speicherort für `/tmp` zu ersetzen, wenn dies in den folgenden Schritten angegeben ist.)

Wählen Sie eine der folgenden beiden Optionen, um die WordPress Quelldateien auf Ihren Entwicklungscomputer zu kopieren. Bei der ersten Option werden integrierte Befehlszeilenaufrufe genutzt. Bei der zweiten Option wird Git verwendet.

Themen

- Um eine Kopie des WordPress Quellcodes zu erhalten (integrierte Befehlszeilenaufrufe)
- Um eine Kopie des WordPress Quellcodes zu erhalten (Git)

Um eine Kopie des WordPress Quellcodes zu erhalten (integrierte Befehlszeilenaufrufe)

1. Rufen Sie den wget Befehl auf, um eine Kopie des WordPress Quellcodes als ZIP-Datei in das aktuelle Verzeichnis herunterzuladen:

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

2. Rufen Sie die Befehle unzip, mkdir, cp und rm auf, um:

   - Entpacken Sie die `master.zip` Datei in das `/tmp/WordPress_Temp` Verzeichnis (Ordner).
   - den entpackten Inhalt in den `/tmp/WordPress`-Zielordner zu kopieren.
   - den temporären `/tmp/WordPress_Temp`-Ordner und die `master`-Datei zu löschen.

   Führen Sie die Befehle einzeln aus:

   ```
   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
```

Dadurch haben Sie einen sauberen Satz von WordPress Quellcodedateien im `/tmp/WordPress` Ordner.

Um eine Kopie des WordPress Quellcodes zu erhalten (Git)

1. Laden Sie Git auf Ihren Entwicklungscomputer herunter und installieren Sie das Programm.
2. Rufen Sie im Ordner `/tmp/WordPress` den Befehl git init auf.
3. Rufen Sie den git clone Befehl zum Klonen des öffentlichen WordPress Repositorys auf und erstellen Sie Ihre eigene Kopie davon im `/tmp/WordPress` Zielordner:

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

   Dadurch erhalten Sie einen sauberen Satz von WordPress Quellcodedateien im `/tmp/WordPress` Ordner.

Erstellen Sie Skripts, um Ihre Anwendung auszuführen

Erstellen Sie als Nächstes einen Ordner und Skripts im Verzeichnis. CodeDeploy verwendet diese Skripts, um Ihre Anwendungsrevision auf der Amazon EC2 EC2-Zielinstanz einzurichten und bereitzustellen. Sie können zum Erstellen der Skripts jeden beliebigen Texteditor verwenden.

1. Erstellen Sie ein Skriptverzeichnis in Ihrer Kopie des WordPress Quellcodes:

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

2. Erstellen Sie eine `install_dependencies.sh`-Datei in `/tmp/WordPress/scripts`. Fügen Sie der Datei die folgenden Zeilen hinzu. Dieses `install_dependencies.sh`-Skript installiert Apache, MySQL und PHP. Außerdem wird MySQL-Support für PHP hinzugefügt.

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

3. Erstellen Sie eine `start_server.sh`-Datei in `/tmp/WordPress/scripts`. Fügen Sie der Datei die folgenden Zeilen hinzu. Dieses `start_server.sh`-Skript startet Apache und MySQL.

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

4. Erstellen Sie eine `stop_server.sh`-Datei in `/tmp/WordPress/scripts`. Fügen Sie der Datei die folgenden Zeilen hinzu. Dieses `stop_server.sh`-Skript stoppt Apache und 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. Erstellen Sie eine `create_test_db.sh`-Datei in `/tmp/WordPress/scripts`. Fügen Sie der Datei die folgenden Zeilen hinzu. Dieses `create_test_db.sh` Skript verwendet MySQL, um eine **`test`** Datenbank zur Verwendung WordPress zu erstellen.

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

```
mysql -uroot <<CREATE_TEST_DB
CREATE DATABASE IF NOT EXISTS test;
CREATE_TEST_DB
```

6. Erstellen Sie schließlich ein `change_permissions.sh`-Skript in `/tmp/WordPress/scripts`. Damit werden die Ordnerberechtigungen in Apache geändert.

> **Important**
>
> In diesem Skript wurden die Berechtigungen für den `/tmp/WordPress`-Ordner aktualisiert, damit alle Benutzer darin schreiben können. Dies ist erforderlich, damit WordPress währenddessen in die Datenbank geschrieben werden kannSchritt 5: Aktualisieren Sie Ihre WordPress Anwendung und stellen Sie sie erneut bereit. Führen Sie nach der Einrichtung der WordPress Anwendung den folgenden Befehl aus, um die Berechtigungen auf eine sicherere Einstellung zu aktualisieren:
>
> ```
> chmod -R 755 /var/www/html/WordPress
> ```

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

7. Geben Sie allen Skripts ausführbare Berechtigungen. Geben Sie an der Befehlszeile Folgendes ein:

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

Fügen Sie eine Anwendungsspezifikationsdatei hinzu

Fügen Sie als Nächstes eine Anwendungsspezifikationsdatei (Datei) hinzu, eine AppSpec Datei im YAML-Format, die von verwendet wird, um: CodeDeploy

- Ordnen Sie die Quelldateien in Ihrer Anwendungsrevision ihren Zielen auf der Amazon EC2 EC2-Zielinstanz zu.
- Geben Sie benutzerdefinierte Berechtigungen für bereitgestellte Dateien an.

- Geben Sie die Skripts an, die während der Bereitstellung auf der Amazon EC2 EC2-Zielinstanz ausgeführt werden sollen.

Die AppSpec Datei muss benannt `appspec.yml` werden. Sie muss sich im Stammverzeichnis des Anwendungsquellcodes befinden. Das Stammverzeichnis dieses Tutorials ist `/tmp/WordPress`.

Erstellen Sie mit Ihrem Texteditor eine Datei mit dem Namen `appspec.yml`. Fügen Sie der Datei die folgenden Zeilen hinzu:

```
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 verwendet diese AppSpec Datei, um alle Dateien im Ordner auf dem Entwicklungscomputer in den `/tmp/WordPress` Ordner auf der `/var/www/html/WordPress` Amazon EC2 EC2-Zielinstanz zu kopieren. CodeDeploy Führt während der Bereitstellung die angegebenen Skripts wie `root` im `/var/www/html/WordPress/scripts` Ordner auf der Amazon EC2 EC2-Zielinstanz zu bestimmten Ereignissen während des Bereitstellungszyklus aus, z. B. **`BeforeInstall`** und**`AfterInstall`**. Wenn die Ausführung eines dieser Skripts länger als 300

Sekunden (5 Minuten) dauert, wird die Bereitstellung CodeDeploy gestoppt und die Bereitstellung als fehlgeschlagen markiert.

Weitere Informationen zu diesen Einstellungen finden Sie in der CodeDeploy AppSpec Dateiverweis.

> **Important**
>
> Die Platzierung und Anzahl von Leerzeichen zwischen den einzelnen Elementen in dieser Datei ist wichtig. Wenn der Abstand falsch ist, wird CodeDeploy ein Fehler ausgelöst, der möglicherweise schwer zu debuggen ist. Weitere Informationen finden Sie unter AppSpec Abstand zwischen den Dateien.

Schritt 3: Laden Sie Ihre WordPress Bewerbung auf Amazon S3 hoch

Jetzt bereiten Sie Ihre Quellinhalte vor und laden sie an einen Ort hoch, von dem aus Sie sie bereitstellen CodeDeploy können. Die folgenden Anweisungen zeigen Ihnen, wie Sie einen Amazon S3 S3-Bucket bereitstellen, die Dateien der Anwendungsrevision für den Bucket vorbereiten, die Dateien der Revision bündeln und dann die Revision in den Bucket übertragen.

> **Note**
>
> Obwohl dies in diesem Tutorial nicht behandelt wird, können Sie es verwenden, CodeDeploy um Anwendungen von GitHub Repositorys auf Instances bereitzustellen. Weitere Informationen finden Sie unter Integrieren CodeDeploy mit GitHub.

Themen

- Bereitstellen eines Amazon S3 S3-Buckets
- Bereiten Sie die Dateien der Anwendung für den Bucket vor
- Bündeln Sie die Dateien der Anwendung in einer einzigen Archivdatei und übertragen Sie die Archivdatei per Push

Bereitstellen eines Amazon S3 S3-Buckets

Erstellen Sie einen Speichercontainer oder Bucket in Amazon S3 — oder verwenden Sie einen vorhandenen Bucket. Stellen Sie sicher, dass Sie die Revision in den Bucket hochladen können und

dass Amazon EC2 EC2-Instances, die in Bereitstellungen verwendet werden, die Revision aus dem Bucket herunterladen können.

Sie können die AWS CLI, die Amazon S3 S3-Konsole oder die Amazon S3 S3-APIs verwenden, um einen Amazon S3 S3-Bucket zu erstellen. Nachdem Sie den Bucket erstellt haben, stellen Sie sicher, dass Sie Zugriffsberechtigungen für den Bucket und Ihr AWS Konto erteilen.

> **Note**
>
> Bucket-Namen müssen in Amazon S3 für alle AWS Konten eindeutig sein. Falls Sie **`amzn-s3-demo-bucket`** nicht verwenden können: Versuchen Sie es mit einem anderen Bucket-Namen wie **`amzn-s3-demo-bucket`**, gefolgt von einem Bindestrich und Ihren Initialen oder einem anderen eindeutigen Bezeichner. Anschließend müssen Sie überall im Tutorial **`amzn-s3-demo-bucket`** durch Ihren Bucket-Namen ersetzen.
> Der Amazon S3 S3-Bucket muss in derselben AWS Region erstellt werden, in der Ihre Amazon EC2 EC2-Ziel-Instances gestartet werden. Wenn Sie den Bucket beispielsweise in der Region USA Ost (Nord-Virginia) erstellen, müssen Ihre Amazon EC2 EC2-Ziel-Instances in der Region USA Ost (Nord-Virginia) gestartet werden.

Themen

- So erstellen Sie einen Amazon S3 S3-Bucket (CLI)
- So erstellen Sie einen Amazon S3 S3-Bucket (Konsole)
- Erteilen Sie dem Amazon S3 S3-Bucket Berechtigungen und AWS Konto

So erstellen Sie einen Amazon S3 S3-Bucket (CLI)

Rufen Sie den mb Befehl auf, um einen Amazon S3 S3-Bucket mit dem Namen zu erstellen**`amzn-s3-demo-bucket`**:

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

So erstellen Sie einen Amazon S3 S3-Bucket (Konsole)

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Amazon S3 S3-Konsole Create Bucket aus.

3. Geben Sie im Feld Bucket name (Bucket-Name) einen Namen für den Bucket ein.
4. Wählen Sie in der Liste Region die Zielregion aus und wählen Sie dann Create aus.

Erteilen Sie dem Amazon S3 S3-Bucket Berechtigungen und AWS Konto

Sie müssen über Berechtigungen zum Hochladen in den Amazon S3 S3-Bucket verfügen. Sie können diese Berechtigungen über eine Amazon S3 S3-Bucket-Richtlinie angeben. In der folgenden Amazon S3 S3-Bucket-Richtlinie ermöglicht die Verwendung des Platzhalterzeichens (*) dem AWS Konto `111122223333` beispielsweise das Hochladen von Dateien in ein beliebiges Verzeichnis im Amazon S3 S3-Bucket mit dem Namen`amzn-s3-demo-bucket`:

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

Informationen zur Anzeige Ihrer AWS Konto-ID finden Sie unter Ihre AWS Konto-ID finden.

Jetzt ist ein guter Zeitpunkt, um zu überprüfen, ob der Amazon S3 S3-Bucket Download-Anfragen von jeder teilnehmenden Amazon EC2 EC2-Instance zulässt. Sie können dies über eine Amazon S3 S3-Bucket-Richtlinie angeben. In der folgenden Amazon S3 S3-Bucket-Richtlinie ermöglicht die Verwendung des Platzhalterzeichens (*) beispielsweise jeder Amazon EC2 EC2-Instance mit einem angehängten IAM-Instance-Profil, das den ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` enthält, Dateien aus einem beliebigen Verzeichnis im Amazon S3 S3-Bucket mit dem Namen herunterzuladen: `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"
                ]
            }
        }
    ]
}
```

Informationen zum Generieren und Anhängen einer Amazon S3 S3-Bucket-Richtlinie finden Sie unter Beispiele für Bucket-Richtlinien.

Informationen zum Erstellen und Anhängen einer IAM-Richtlinie finden Sie unter Arbeiten mit Richtlinien.

Bereiten Sie die Dateien der Anwendung für den Bucket vor

Stellen Sie sicher, dass die WordPress Anwendungsdateien, die AppSpec Datei und die Skripts auf Ihrem Entwicklungscomputer ähnlich wie folgt organisiert sind:

```
/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...)
```

Bündeln Sie die Dateien der Anwendung in einer einzigen Archivdatei und übertragen Sie die Archivdatei per Push

Bündeln Sie die WordPress Anwendungsdateien und die AppSpec Datei in einer Archivdatei (bekannt als Anwendungsrevision).

> **Note**
>
> Möglicherweise werden Ihnen für die Speicherung von Objekten in einem Bucket und die Übertragung von Anwendungsrevisionen in einen oder aus einem Bucket Gebühren berechnet. Weitere Informationen finden Sie unter Amazon S3 – Preise.

1. Wechseln Sie auf dem Entwicklungscomputer in den Ordner, in dem die Dateien abgelegt sind:

   ```
   cd /tmp/WordPress
   ```

   > **Note**
   >
   > Wenn Sie nicht in diesen Ordner wechseln, startet die Dateibündelung im jeweils aktuellen Ordner. Beispiel: Wenn der aktuelle Ordner `/tmp` statt `/tmp/WordPress` ist, beginnt der Bündelvorgang mit den Dateien und Unterordnern im Ordner `tmp`, in dem u. U. mehr enthalten ist als bloß die `WordPress`-Unterordner.

2. Rufen Sie den Befehl create-application auf, um eine neue Anwendung namens **`WordPress_App`** zu registrieren:

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

3. Rufen Sie den CodeDeploy Push-Befehl auf, um die Dateien zu bündeln, die Revisionen auf Amazon S3 hochzuladen und Informationen CodeDeploy über die hochgeladene Version zu registrieren — alles in einer Aktion.

   ```
   aws deploy push \
   ```

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

Dieser Befehl bündelt die Dateien aus dem aktuellen Verzeichnis (mit Ausnahme aller versteckten Dateien) in einer einzigen Archivdatei mit dem Namen**`WordPressApp.zip`**, lädt die Revision in den **`amzn-s3-demo-bucket`** Bucket hoch und registriert Informationen CodeDeploy über die hochgeladene Revision.

Schritt 4: Stellen Sie Ihre WordPress Anwendung bereit

Jetzt stellen Sie die Version der WordPress Beispielanwendung bereit, die Sie auf Amazon S3 hochgeladen haben. Sie können die AWS CLI oder die CodeDeploy Konsole verwenden, um die Version bereitzustellen und den Fortschritt der Bereitstellung zu überwachen. Im Anschluss an die Bereitstellung der Anwendungsrevision überprüfen Sie die Ergebnisse.

Themen

- Stellen Sie die Revision Ihrer Anwendung bereit mit CodeDeploy
- Überwachen Sie Ihre Bereitstellung und beheben Sie Fehler
- Überprüfen Sie Ihre Bereitstellung

Stellen Sie die Revision Ihrer Anwendung bereit mit CodeDeploy

Verwenden Sie die Konsole AWS CLI oder die Konsole, um Ihre Anwendungsrevision bereitzustellen.

Themen

- Mit der CLI stellen Sie die Anwendungsrevision wie folgt bereit:
- Mit der Konsole stellen Sie die Anwendungsrevision wie folgt bereit:

Mit der CLI stellen Sie die Anwendungsrevision wie folgt bereit:

1. Für die Bereitstellung ist eine Bereitstellungsgruppe erforderlich. Bevor Sie die Bereitstellungsgruppe erstellen können, benötigen Sie allerdings den ARN einer Service-Rolle. Eine Servicerolle ist eine IAM-Rolle, die einem Dienst die Erlaubnis erteilt, in Ihrem Namen zu handeln. In diesem Fall erteilt die Service-Rolle die CodeDeploy Erlaubnis, auf Ihre Amazon EC2 EC2-Instances zuzugreifen, um deren Amazon EC2 EC2-Instance-Tags zu erweitern (zu lesen).

Sie sollten bereits anhand der Anweisungen unter Erstellen Sie eine Servicerolle (CLI) eine Service-Rolle erstellt haben. Wie Sie den ARN der Service-Rolle abrufen können, erfahren Sie unter Rufen Sie die Servicerolle ARN (CLI) ab.

2. Nachdem Sie die Dienstrolle ARN haben, rufen Sie den create-deployment-group Befehl auf, um eine Bereitstellungsgruppe mit dem Namen zu erstellen**`WordPress_DepGroup`**, die der genannten Anwendung zugeordnet ist. Verwenden Sie dazu das Amazon EC2-Tag mit dem Namen **`CodeDeployDemo`** und die Bereitstellungskonfiguration mit dem Namen**`CodeDeployDefault.OneAtATime`**: **`WordPress_App`**

```
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**
>
> Der Befehl create-deployment-group bietet Unterstützung für die Erstellung von Triggern, die zum Senden von Amazon SNS SNS-Benachrichtigungen an Themenabonnenten über bestimmte Ereignisse in Bereitstellungen und Instances führen. Der Befehl unterstützt auch Optionen für das automatische Rollback von Bereitstellungen und die Einrichtung von Alarmen, um Bereitstellungen zu beenden, wenn die Überwachungsschwellenwerte in CloudWatch Amazon-Alarmen erreicht werden. Befehle für diese Aktionen sind in diesem Tutorial nicht enthalten.

3. Bevor Sie eine Bereitstellung erstellen, muss der CodeDeploy Agent auf den Instances in Ihrer Bereitstellungsgruppe installiert sein. Sie können den Agenten über die Befehlszeile mit folgendem Befehl mit AWS Systems Manager installieren:

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

Dieser Befehl erstellt eine Verknüpfung in Systems Manager State Manager, die den CodeDeploy Agenten installiert und dann versucht, ihn jeden Sonntagmorgen um 2:00 Uhr zu aktualisieren. Weitere Informationen zum CodeDeploy Agenten finden Sie unter Mit dem CodeDeploy Agenten arbeiten. Weitere Informationen zu Systems Manager finden Sie unter Was ist AWS Systems Manager.

4. Rufen Sie nun den Befehl create-deployment zum Erstellen einer Bereitstellung auf, die der Anwendung **`WordPress_App`**, der Bereitstellungskonfiguration **`CodeDeployDefault.OneAtATime`** und der Bereitstellungsgruppe **`WordPress_DepGroup`** zugeordnet ist, unter Verwendung der Anwendungrevision **`WordPressApp.zip`** im Bucket **`amzn-s3-demo-bucket`**:

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

Mit der Konsole stellen Sie die Anwendungsrevision wie folgt bereit:

1. Bevor Sie die CodeDeploy Konsole zum Bereitstellen Ihrer Anwendungsrevision verwenden, benötigen Sie einen ARN für die Servicerolle. Eine Servicerolle ist eine IAM-Rolle, die einem Dienst die Erlaubnis erteilt, in Ihrem Namen zu handeln. In diesem Fall erteilt die Service-Rolle die CodeDeploy Erlaubnis, auf Ihre Amazon EC2 EC2-Instances zuzugreifen, um deren Amazon EC2 EC2-Instance-Tags zu erweitern (zu lesen).

   Sie sollten bereits anhand der Anweisungen unter Erstellen Sie eine Servicerolle (Konsole) eine Service-Rolle erstellt haben. Wie Sie den ARN der Service-Rolle abrufen können, erfahren Sie unter Rufen Sie die Servicerolle ARN ab (Konsole).

2. Nachdem Sie den ARN haben, verwenden Sie die CodeDeploy Konsole, um Ihre Anwendungsrevision bereitzustellen:

   Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

3. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
4. Wählen Sie in der Liste der Anwendungen WordPress_App aus.
5. Wählen Sie auf der Registerkarte Deployment groups (Bereitstellungsgruppen), die Option Create deployment group (Bereitstellungsgruppe erstellen) aus.
6. Geben Sie unter Deployment group name (Name der Bereitstellungsgruppe) **`WordPress_DepGroup`** ein.
7. Wählen Sie unter Bereitstellungstyp die Option In-placeBereitstellung aus.
8. Wählen Sie unter Umgebungskonfiguration Amazon EC2 EC2-Instances aus.
9. Behalten Sie unter Agentenkonfiguration mit AWS Systems Manager die Standardeinstellungen bei.
10. Geben Sie für Key (Schlüssel) **`Name`** ein.
11. Geben Sie unter Value (Wert) **`CodeDeployDemo`** ein.

> **Note**
>
> Nach der Eingabe sollte unter Passende Instances eine 1 erscheinen**`CodeDeployDemo`**, um zu bestätigen, dass eine passende Amazon EC2 EC2-Instance CodeDeploy gefunden wurde.

12. Wählen Sie unter Deployment configuration (Bereitstellungskonfiguration) die Option CodeDeployDefault.OneAtATime aus.
13. Wählen Sie in der Dropdownliste Service role ARN (Servicerolle-ARN) den ARN der Servicerolle aus und klicken Sie dann auf Create deployment group (Bereitstellungsgruppe erstellen).
14. Wählen Sie Create deployment.
15. Wählen Sie unter Deployment group (Bereitstellungsgruppe) die Option **`WordPress_DepGroup`** aus.
16. Wählen Sie neben Repository-Typ die Option Meine Anwendung ist in Amazon S3 gespeichert aus. Geben Sie unter Revisionsort den Speicherort der Version der WordPress

Beispielanwendung ein, die Sie zuvor auf Amazon S3 hochgeladen haben. Ermitteln Sie den Speicherort wie folgt:

a. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

b. Wählen Sie in der Liste der Buckets amzn-s3-demo-bucket aus (oder den Namen des Buckets, in den Sie Ihre Anwendungsrevision hochgeladen haben).

c. Wählen Sie aus der Objektliste WordPressApp.zip aus.

d. Kopieren Sie auf der Registerkarte Overview (Übersicht) den Wert des Feldes Link (Link) in die Zwischenablage.

   Er kann wie folgt aussehen:

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

e. Kehren Sie zur CodeDeploy Konsole zurück und fügen Sie in Revisionsort den Feldwert Link ein.

17. Falls in der Liste File type (Dateityp) die Meldung angezeigt wird, dass der Dateityp nicht erkannt wurde, wählen Sie .zip aus der Liste der Dateitypen aus.
18. (Optional) Geben Sie einen Kommentar in dem Feld Deployment description ein.
19. Erweitern Sie Deployment group overrides (Überschreibungen der Bereitstellungsgruppe) und wählen Sie unter Deployment configuration (Bereitstellungskonfiguration) die Option CodeDeployDefault.OneAtATime aus.
20. Klicken Sie auf Start deployment (Bereitstellung starten). Auf der Seite Deployments werden nun Informationen zu der neu erstellten Bereitstellung angezeigt.

Überwachen Sie Ihre Bereitstellung und beheben Sie Fehler

Verwenden Sie die Konsole AWS CLI oder die Konsole, um Ihre Bereitstellung zu überwachen und Fehler zu beheben.

Themen

- Über die CLI können Sie die Bereitstellung wie folgt überwachen und Probleme beheben:
- Über die Konsole können Sie die Bereitstellung wie folgt überwachen und Probleme beheben:

Über die CLI können Sie die Bereitstellung wie folgt überwachen und Probleme beheben:

1. Fordern Sie die ID der Bereitstellung an, indem Sie den Befehl list-deployments für die Anwendung **`WordPress_App`** und die Bereitstellungsgruppe **`WordPress_DepGroup`** aufrufen:

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

2. Rufen Sie den Befehl get-deployment mit der Bereitstellungs-ID auf:

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

3. Der Befehl gibt den allgemeinen Status der Bereitstellung zurück. Wurde die Bereitstellung erfolgreich abgeschlossen, lautet der Wert `Succeeded`.

   Wenn der Gesamtstatus lautet`Failed`, können Sie Befehle wie list-deployment-instances und get-deployment-instance aufrufen, um Fehler zu beheben. Weitere Optionen für die Problembehebung finden Sie unter Analysieren von Protokolldateien zur Untersuchung von Bereitstellungsfehlern auf Instances.

Über die Konsole können Sie die Bereitstellung wie folgt überwachen und Probleme beheben:

Auf der Seite Bereitstellungen in der CodeDeploy Konsole können Sie den Status Ihrer Bereitstellung in der Spalte Status überwachen.

Weitere Informationen zu Ihrer Bereitstellung können Sie wie nachfolgend beschrieben abrufen. Dies ist insbesondere nützlich, falls in der Spalte Status ein anderer Wert als Succeeded steht:

1. Wählen Sie in der Tabelle Deployments (Bereitstellungen) den Namen der Bereitstellung aus. Falls eine Bereitstellung fehlschlägt, wird eine Meldung mit dem Grund des Fehlschlagens angezeigt.
2. Unter Instance activity (Instance-Aktivität) werden weitere Informationen zur Bereitstellung angezeigt. Wenn eine Bereitstellung fehlschlägt, können Sie möglicherweise feststellen, auf welchen Amazon EC2 EC2-Instances und in welchem Schritt die Bereitstellung fehlgeschlagen ist.
3. Für eine weitergehende Problembehebung können Sie wie unter View Instance Details beschrieben vorgehen. Sie können auch die Bereitstellungsprotokolldateien auf einer Amazon

EC2 EC2-Instance analysieren. Weitere Informationen finden Sie unter Analysieren von Protokolldateien zur Untersuchung von Bereitstellungsfehlern auf Instances.

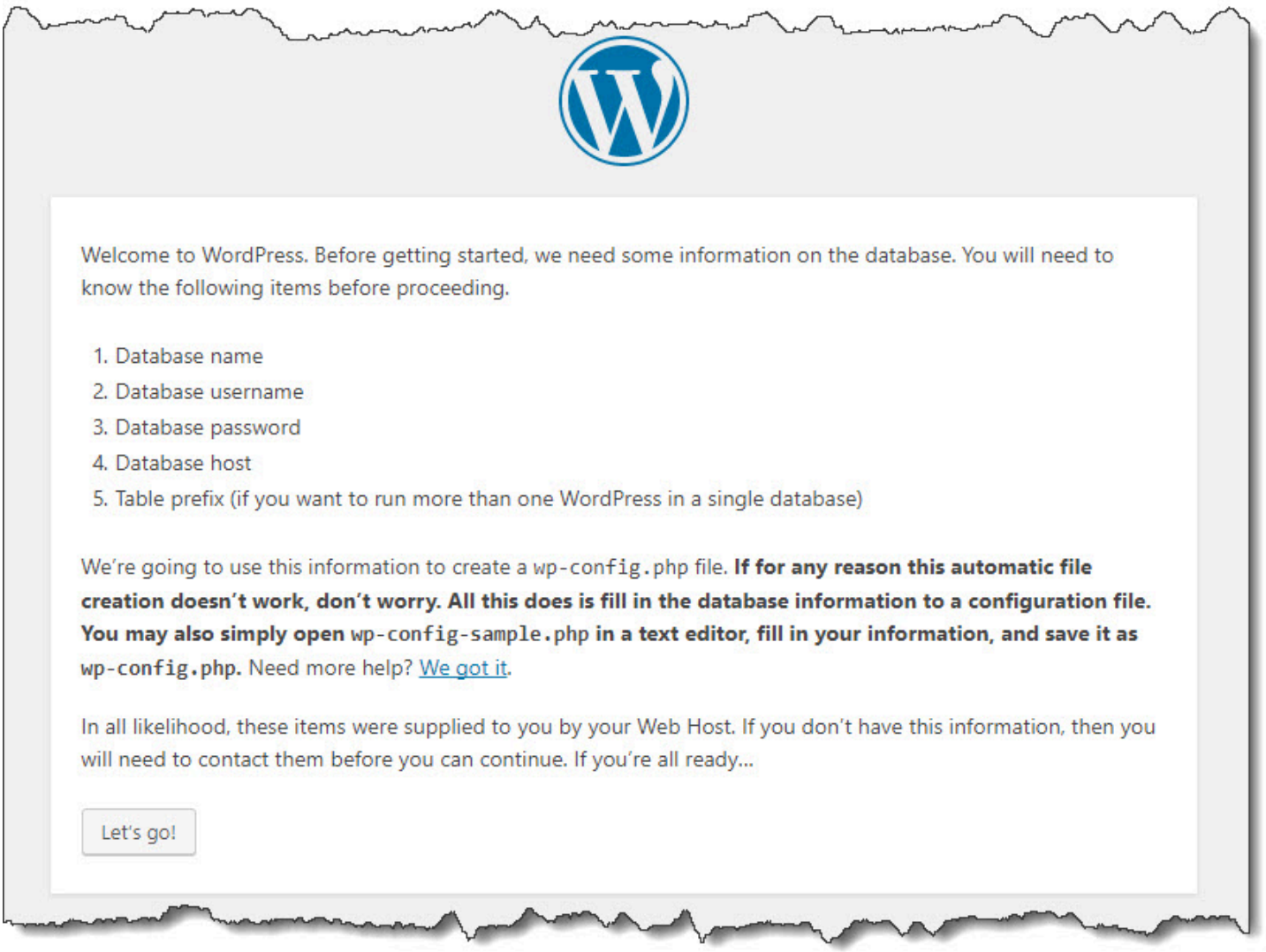
Überprüfen Sie Ihre Bereitstellung

Stellen Sie nach erfolgreicher Bereitstellung sicher, dass Ihre WordPress Installation funktioniert. Verwenden Sie die öffentliche DNS-Adresse der Amazon EC2 EC2-Instance, gefolgt von/`WordPress`, um Ihre Site in einem Webbrowser anzuzeigen. (Um den öffentlichen DNS-Wert abzurufen, wählen Sie in der Amazon EC2 EC2-Konsole die Amazon EC2 EC2-Instance aus und suchen Sie auf der Registerkarte Beschreibung nach dem Wert von Public DNS.)

Wenn die öffentliche DNS-Adresse Ihrer Amazon EC2 EC2-Instance beispielsweise lautet**`ec2-01-234-567-890.compute-1.amazonaws.com`**, würden Sie die folgende URL verwenden:

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

Wenn Sie die Website in Ihrem Browser aufrufen, sollten Sie eine WordPress Willkommensseite sehen, die der folgenden ähnelt:



Wenn Ihrer Amazon EC2 EC2-Instance keine Regel für eingehenden HTTP-Datenverkehr zur Sicherheitsgruppe hinzugefügt wurde, wird die WordPress Willkommensseite nicht angezeigt. Wenn Sie eine Meldung sehen, die besagt, dass der Remote-Server nicht reagiert, stellen Sie sicher, dass die Sicherheitsgruppe für Ihre Amazon EC2 EC2-Instance die eingehende Regel hat. Weitere Informationen finden Sie unter Fügen Sie eine Regel für eingehenden Datenverkehr hinzu, die HTTP-Verkehr zu Ihrer Amazon Linux- oder RHEL Amazon EC2 EC2-Instance zulässt.

Schritt 5: Aktualisieren Sie Ihre WordPress Anwendung und stellen Sie sie erneut bereit

Nachdem Sie Ihre Anwendungsversion erfolgreich bereitgestellt haben, aktualisieren Sie den WordPress Code auf dem Entwicklungscomputer und verwenden Sie ihn dann, CodeDeploy um

die Site erneut bereitzustellen. Danach sollten Sie die Codeänderungen auf der Amazon EC2 EC2-Instance sehen.

Themen

- Richten Sie die Site ein WordPress
- Ändern Sie die Site
- Stellen Sie die Site erneut bereit

Richten Sie die Site ein WordPress

Um die Auswirkungen der Codeänderung zu sehen, beenden Sie die Einrichtung der WordPress Site, sodass Sie über eine voll funktionsfähige Installation verfügen.

1. Geben Sie die URL Ihrer Site in Ihren Webbrowser ein. Die URL ist die öffentliche DNS-Adresse der Amazon EC2 EC2-Instance plus eine `/WordPress` Erweiterung. Für diese WordPress Beispielseite (und die öffentliche DNS-Adresse der Amazon EC2 EC2-Instance) lautet die URL. **`http://ec2-01-234-567-890.compute-1.amazonaws.com/WordPress`**
2. Wenn Sie die Site noch nicht eingerichtet haben, wird die WordPress Standard-Willkommensseite angezeigt. Wählen Sie Let's go! (Los geht's!).
3. Geben Sie auf der Seite Datenbankkonfiguration die folgenden Werte ein, um die Standard-MySQL-Datenbank zu nutzen:

   - Datenbankname: **`test`**
   - Benutzername: **`root`**
   - Passwort: Lassen Sie das Feld leer.
   - Datenbank-Host: **`localhost`**
   - Tabellenpräfix: **`wp_`**

   Wählen Sie Submit, um die Datenbank einzurichten.
4. Fahren Sie mit der Einrichtung der Website fort. Geben Sie auf der Willkommensseite alle gewünschten Werte ein und wählen Sie Installieren WordPress. Wenn die Installation abgeschlossen ist, können Sie sich bei Ihrem Dashboard anmelden.

> **Important**
>
> Während der Bereitstellung der WordPress Anwendung hat das **`change_permissions.sh`** Skript die Berechtigungen für den `/tmp/WordPress` Ordner aktualisiert, sodass jeder in den Ordner schreiben kann. Nun ist eine gute Gelegenheit, den folgenden Befehl auszuführen, um die Berechtigungen so einzuschränken, dass nur Sie als Eigentümer darin schreiben können:
>
> ```
> chmod -R 755 /var/www/html/WordPress
> ```

Ändern Sie die Site

Um die WordPress Site zu ändern, wechseln Sie in den Ordner der Anwendung auf Ihrem Entwicklungscomputer:

```
cd /tmp/WordPress
```

Verwenden Sie zur Modifizierung einiger der Farben der Website in der Datei `wp-content/themes/twentyfifteen/style.css` einen Texteditor oder sed, um `#fff` in `#768331` zu ändern.

Bei Linux oder anderen Systemen mit GNU sed verwenden Sie:

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

Bei macOS, Unix oder anderen Systemen mit BSD sed verwenden Sie:

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

Stellen Sie die Site erneut bereit

Nachdem Sie den Code der Site geändert haben, verwenden Sie Amazon S3 und CodeDeploy stellen Sie die Site erneut bereit.

Bündeln Sie die Änderungen und laden Sie sie auf Amazon S3 hoch, wie unter beschriebenBündeln Sie die Dateien der Anwendung in einer einzigen Archivdatei und übertragen Sie die Archivdatei per

Push. (Wenn Sie diese Anweisungen befolgen, müssen Sie keine Anwendung erstellen.) Geben Sie der neuen Revision denselben Schlüssel wie zuvor (**`WordPressApp.zip`**). Laden Sie es in denselben Amazon S3 S3-Bucket hoch, den Sie zuvor erstellt haben (z. B.**`amzn-s3-demo-bucket`**).

Verwenden Sie die AWS CLI, die CodeDeploy Konsole oder die CodeDeploy APIs, um die Site erneut bereitzustellen.

Themen

- Erneutes Bereitstellen der Website (CLI)
- Erneutes Bereitstellen der Website (Konsole)

Erneutes Bereitstellen der Website (CLI)

Rufen Sie den Befehl create-deployment auf, um eine auf der neu hochgeladenen Revision basierende Bereitstellung zu erstellen. Verwenden Sie die Anwendung mit dem Namen **`WordPress_App`**, die Bereitstellungskonfiguration mit dem Namen **`CodeDeployDefault.OneAtATime`**, die Bereitstellungsgruppe mit dem Namen **`WordPress_DepGroup`** und die Revision mit dem Namen **`WordPressApp.zip`** im Bucket mit dem Namen **`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
```

Sie können den Status der Bereitstellung wie in Überwachen Sie Ihre Bereitstellung und beheben Sie Fehler beschrieben überprüfen.

Rufen Sie die Site nach CodeDeploy der erneuten Bereitstellung erneut in Ihrem Webbrowser auf, um zu überprüfen, ob die Farben geändert wurden. (Sie müssen möglicherweise Ihren Browser aktualisieren.) Wenn die Farben geändert wurden, herzlichen Glückwunsch! Sie haben Ihre Website erfolgreich geändert und erneut bereitgestellt!

Erneutes Bereitstellen der Website (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter. https://console.aws.amazon.com/codedeploy

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie in der Liste der Anwendungen WordPress_App aus.
4. Wählen Sie auf der Registerkarte Deployment groups (Bereitstellungsgruppen) die Option **`WordPress_DepGroup`**.
5. Wählen Sie Create deployment.
6. Auf der Seite Create deployment:

   a. Wählen Sie unter Deployment group (Bereitstellungsgruppe) die Option **`WordPress_DepGroup`** aus.

   b. Wählen Sie im Bereich Repository-Typ die Option Meine Anwendung ist in Amazon S3 gespeichert aus und kopieren Sie dann den Amazon S3 S3-Link Ihrer Revision in das Feld Speicherort der Revision. Ermitteln des Link-Wertes:

      i. In einer separaten Browser-Registerkarte:

         Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

         Navigieren Sie zu amzn-s3-demo-bucket, öffnen Sie es und wählen Sie dann Ihre Revision aus. **`WordPressApp.zip`**

      ii. Wenn der Eigenschaftenbereich in der Amazon S3 S3-Konsole nicht sichtbar ist, wählen Sie die Schaltfläche Eigenschaften.

      iii. Kopieren Sie im Eigenschaftenbereich den Wert des Link-Felds in das Feld Revisionsort in der CodeDeploy Konsole.

   c. Wenn eine Meldung angezeigt wird, die besagt, dass der Dateityp nicht erkannt werden konnte, wählen Sie .zip.

   d. Lassen Sie das Feld Deployment description leer.

e. Erweitern Sie Deployment group overrides (Überschreibungen der Bereitstellungsgruppe) und wählen Sie unter Deployment configuration (Bereitstellungskonfiguration) die Option CodeDeployDefault.OneAtATime.

f. Klicken Sie auf Start deployment (Bereitstellung starten). Auf der Seite Deployments werden nun Informationen zu der neu erstellten Bereitstellung angezeigt.

g. Sie können den Status der Bereitstellung wie in Überwachen Sie Ihre Bereitstellung und beheben Sie Fehler beschrieben überprüfen.

   Rufen Sie die Site nach CodeDeploy der erneuten Bereitstellung erneut in Ihrem Webbrowser auf, um zu überprüfen, ob die Farben geändert wurden. (Sie müssen möglicherweise Ihren Browser aktualisieren.) Wenn die Farben geändert wurden, herzlichen Glückwunsch! Sie haben Ihre Website erfolgreich geändert und erneut bereitgestellt!

Schritt 6: Bereinigen Sie Ihre WordPress Anwendung und die zugehörigen Ressourcen

Sie haben den WordPress Code jetzt erfolgreich aktualisiert und die Site erneut bereitgestellt. Um laufende Gebühren für die Ressourcen, die Sie für dieses Tutorial erstellt haben, zu vermeiden, sollten Sie Folgendes löschen:

- Alle CloudFormation Stacks (oder beenden Sie alle Amazon EC2 EC2-Instances, falls Sie sie außerhalb von CloudFormation erstellt haben).
- Alle Amazon S3 S3-Buckets.
- Die Anwendung `WordPress_App` in CodeDeploy.
- Die AWS Systems Manager State Manager-Zuordnung für den CodeDeploy Agenten.

Sie können Amazon S3 AWS CLI CloudFormation, Amazon EC2 und CodeDeploy Konsolen oder die AWS APIs verwenden, um die Bereinigung durchzuführen.

Themen

- So bereinigen Sie die Ressourcen (CLI):
- So bereinigen Sie Ressourcen (Konsole):
- Als nächstes

So bereinigen Sie die Ressourcen (CLI):

1. Wenn Sie unsere CloudFormation Vorlage für dieses Tutorial verwendet haben, rufen Sie den delete-stack Befehl für den genannten Stack auf. **`CodeDeployDemoStack`** Dadurch werden alle zugehörigen Amazon EC2 EC2-Instances beendet und alle zugehörigen IAM-Rollen gelöscht, die der Stack erstellt hat:

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

2. Um den Amazon S3 S3-Bucket zu löschen, rufen Sie den rm Befehl mit dem --recursive Schalter für den genannten Bucket auf**`amzn-s3-demo-bucket`**. So löschen Sie den Bucket und alle Objekte im Bucket:

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

3. Um die Anwendung `WordPress_App` zu löschen, rufen Sie den Befehl delete-application auf. Damit werden auch alle zugehörigen Bereitstellungsgruppen-Datensätze und die Bereitstellungsdatensätze für die Anwendung gelöscht:

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

4. Rufen Sie den delete-association Befehl auf, um die Systems Manager State Manager-Zuordnung zu löschen.

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

   Sie können den abrufen, *`association-id`* indem Sie den describe-association Befehl aufrufen.

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

Wenn Sie den CloudFormation Stack für dieses Tutorial nicht verwendet haben, rufen Sie den terminate-instances Befehl auf, um alle Amazon EC2 EC2-Instances zu beenden, die Sie manuell erstellt haben. Geben Sie die ID der Amazon EC2 EC2-Instance an, die beendet werden soll:

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

So bereinigen Sie Ressourcen (Konsole):

Wenn Sie unsere CloudFormation Vorlage für dieses Tutorial verwendet haben, löschen Sie den zugehörigen CloudFormation Stack.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Geben Sie im Feld Filter den CloudFormation Stacknamen ein, den Sie zuvor erstellt haben (z. B.**`CodeDeployDemoStack`**).
3. Aktivieren Sie das Kontrollkästchen neben dem Stack-Namen. Wählen Sie im Menü Actions die Option Delete Stack.

   CloudFormation löscht den Stack, beendet alle zugehörigen Amazon EC2 EC2-Instances und löscht alle zugehörigen IAM-Rollen.

So beenden Sie Amazon EC2 EC2-Instances, die Sie außerhalb eines CloudFormation Stacks erstellt haben:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der Liste INSTANCES die Option Instances.
3. Geben Sie in das Suchfeld den Namen der Amazon EC2 EC2-Instance ein, die Sie beenden möchten (z. B.**`CodeDeployDemo`**), und drücken Sie dann die Eingabetaste.
4. Wählen Sie den Namen der Amazon EC2 EC2-Instanz.
5. Zeigen Sie im Menü Actions auf Instance State, und wählen Sie dann Terminate. Wählen Sie bei Aufforderung Yes, Terminate.

Wiederholen Sie diese Schritte für jede Instance.

Um den Amazon S3 S3-Bucket zu löschen:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Suchen Sie in der Liste der Buckets nach dem Namen des Amazon S3 S3-Buckets, den Sie zuvor erstellt haben, und wählen Sie ihn aus (z. B.**`amzn-s3-demo-bucket`**).

3. Bevor Sie einen Bucket löschen können, müssen Sie zunächst seinen Inhalt löschen. Wählen Sie alle Dateien im Bucket aus, z. B. **`WordPressApp.zip`**. Wählen Sie im Menü Actions die Option Delete. Wenn Sie zum Bestätigen des Löschvorgangs aufgefordert werden, wählen Sie OK.
4. Nachdem der Bucket leer ist, kann er gelöscht werden. Wählen Sie in der Bucket-Liste die Zeile mit dem Bucket aus (aber nicht den Namen des Buckets). Wählen Sie Delete bucket und bei der Aufforderung zum Bestätigen OK.

Um die `WordPress_App` Anwendung zu löschen von CodeDeploy:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie in der Liste der Anwendungen die Option WordPress_App aus.
4. Wählen Sie auf der Seite Application details die Option Delete application.
5. Geben Sie bei Aufforderung den Namen der Anwendung ein, um das Löschen zu bestätigen, und klicken Sie auf Delete (Löschen).

So löschen Sie die Systems Manager State Manager-Zuordnung:

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager.
2. Wählen Sie im Navigationsbereich Statusmanager aus.
3. Wählen Sie die Verknüpfung aus, die Sie erstellt haben, und wählen Sie Delete (Löschen) aus.

Als nächstes

Glückwunsch, dass Sie hier angekommen sind! Sie haben erfolgreich eine CodeDeploy-Bereitstellung abgeschlossen, dann den Code Ihrer Site aktualisiert und ihn erneut bereitgestellt.

Tutorial: Stellen Sie ein „Hallo, Welt!“ bereit Anwendung mit CodeDeploy (Windows Server)

In diesem Tutorial stellen Sie eine einzelne Webseite auf einer einzelnen Windows Server-Amazon EC2-Instance bereit, auf der Internet Information Services (IIS) als Webserver ausgeführt wird. Auf dieser Webseite wird ein einfaches „Hallo, Welt!“ angezeigt message.

Nicht das, wonach Sie gesucht haben?

- Um stattdessen die Bereitstellung auf einer Amazon Linux- oder Red Hat Enterprise Linux (RHEL) Amazon EC2 EC2-Instance zu üben, siehe. Tutorial: Bereitstellung WordPress auf einer Amazon EC2 EC2-Instance (Amazon Linux oder Red Hat Enterprise Linux und Linux, macOS oder Unix)
- Informationen zum Bereitstellen auf einer lokalen Instance finden Sie unter Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat Enterprise Linux).

Die Schritte in diesem Tutorial gelten für Windows. Obwohl Sie die meisten dieser Schritte auf einem lokalen Computer unter Linux, macOS oder Unix ausführen können, müssen Sie diejenigen anpassen, die Windows-based Verzeichnispfade abdecken, wie `c:\temp` z. Wenn Sie eine Verbindung zur Amazon EC2 EC2-Instance herstellen möchten, benötigen Sie außerdem eine Client-Anwendung, die über das Remote Desktop Protocol (RDP) eine Verbindung zur Amazon EC2 EC2-Instance herstellen kann, auf der Windows Server ausgeführt wird. (Windows bietet standardmäßig eine RDP-Verbindungs-Client-Anwendung.)

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die Voraussetzungen unter erfüllenErste Schritte mit CodeDeploy, einschließlich der Konfiguration Ihres Benutzers, der Installation oder Aktualisierung von und der AWS CLI Erstellung eines IAM-Instance-Profils und einer Servicerolle.

Themen

- Schritt 1: Starten Sie eine Windows Server Amazon EC2 EC2-Instance
- Schritt 2: Konfigurieren Sie Ihren Quellinhalt für die Bereitstellung auf der Windows Server Amazon EC2 EC2-Instance
- Schritt 3: Lade dein „Hallo, Welt!“ hoch Anwendung auf Amazon S3
- Schritt 4: Stellen Sie Ihre Hello World-Anwendung bereit
- Schritt 5: Aktualisieren Sie Ihr „Hallo, Welt!“ und stellen Sie es erneut bereit Anwendung
- Schritt 6: Räumen Sie Ihr „Hallo, Welt!“ auf Anwendung und zugehörige Ressourcen

Schritt 1: Starten Sie eine Windows Server Amazon EC2 EC2-Instance

Um die Hello World-Anwendung mit bereitzustellen CodeDeploy, benötigen Sie eine Amazon EC2 EC2-Instance, auf der Windows Server ausgeführt wird.

Folgen Sie den Anweisungen in Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy. Wenn Sie bereit sind, der Instance ein Amazon EC2 EC2-Instance-Tag zuzuweisen, geben Sie unbedingt den Tag-Schlüssel von **`Name`** und den Tag-Wert von **`CodeDeployDemo`** an. (Wenn Sie einen anderen Tag-Schlüssel oder Tag-Wert angeben, dann könnten die Anweisungen in Schritt 4: Stellen Sie Ihre Hello World-Anwendung bereit zu unerwarteten Ergebnissen führen.)

Nachdem Sie die Amazon EC2 EC2-Instance gestartet haben, kehren Sie zu dieser Seite zurück und fahren Sie mit dem nächsten Abschnitt fort. Fahren Sie nicht mit Erstellen Sie eine Anwendung mit CodeDeploy als nächsten Schritt fort.

Connect zu Ihrer Amazon EC2 EC2-Instance her

Nachdem Ihre Amazon EC2 EC2-Instance gestartet wurde, folgen Sie diesen Anweisungen, um zu üben, wie Sie sich mit ihr verbinden.

> **Note**
>
> In dieser Anleitung wird davon ausgegangen, dass Windows und die Windows Desktop Connection-Clientanwendung ausgeführt werden. Weitere Informationen finden Sie unter Herstellen einer Verbindung zu Ihrer Windows-Instance mithilfe von RDP. Unter Umständen müssen Sie diese Anweisungen für andere Betriebssysteme oder andere Clientanwendungen für RDP-Verbindungen anpassen.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances die Option Instances.
3. Navigieren Sie zu Ihrer Windows Server-Instance und wählen Sie sie in der Liste aus.
4. Wählen Sie Connect aus.
5. Wählen Sie Get Password (Passwort abrufen) und wählen Sie dann Choose File (Datei wählen).
6. Suchen Sie die Amazon EC2-Instance-Schlüsselpaardatei, die mit der Amazon EC2-Instance von Windows Server verknüpft ist, wählen Sie sie aus, und wählen Sie dann Öffnen aus.

7. Klicken Sie auf Decrypt Password. Notieren Sie sich das angezeigte Passwort. Sie benötigen ihn in Schritt 10.
8. Wählen Sie Download Remote Desktop File, und öffnen Sie dann die Datei.
9. Wenn Sie zum Verbinden aufgefordert werden, obwohl der Herausgeber der Remote-Verbindung nicht identifiziert werden kann, fahren Sie fort.
10. Geben Sie das Passwort ein, das Sie sich in Schritt 7 notiert haben, und fahren Sie dann fort. (Wenn Ihre Clientanwendung für die RDP-Verbindung Sie zur Eingabe eines Benutzernamens auffordert, geben Sie **`Administrator`** ein.)
11. Wenn Sie zum Verbinden aufgefordert werden, obwohl die Identität des Remotecomputers nicht überprüft werden konnte, fahren Sie fort.
12. Nachdem Sie eine Verbindung hergestellt haben, wird der Desktop der Amazon EC2 EC2-Instance angezeigt, auf der Windows Server ausgeführt wird.
13. Sie können jetzt die Verbindung zur Amazon EC2 EC2-Instance trennen.

> **Warning**
>
> Stoppen oder beenden Sie die Instance nicht. Andernfalls CodeDeploy kann es nicht darauf bereitgestellt werden.

Fügen Sie eine Regel für eingehenden Datenverkehr hinzu, die HTTP-Verkehr zu Ihrer Windows Server Amazon EC2 EC2-Instance zulässt

Im nächsten Schritt wird bestätigt, dass Ihre Amazon EC2 EC2-Instance über einen offenen HTTP-Port verfügt, sodass Sie die bereitgestellte Webseite auf Ihrer Windows Server Amazon EC2 EC2-Instance in einem Browser sehen können.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie Instances und dann Ihre Instance aus.
3. Wählen Sie auf der Registerkarte Beschreibung unter Sicherheitsgruppen die Option Regeln für eingehenden Datenverkehr anzeigen aus.

   In Ihrer Sicherheitsgruppe sollte eine Liste mit Regeln wie die folgende angezeigt werden:

```
Security Groups associated with i-1234567890abcdef0
```

```
Ports     Protocol     Source     launch-wizard-N
22        tcp          0.0.0.0/0          #
```

4. Wählen Sie unter Sicherheitsgruppen die Sicherheitsgruppe für Ihre Amazon EC2 EC2-Instance aus. Ein möglicher Name ist **`launch-wizard-N`**. Beim ***N*** im Namen handelt es sich um eine Nummer, die Ihrer Sicherheitsgruppe beim Erstellen Ihrer Instance zugewiesen wurde.

   Wählen Sie die Registerkarte Inbound (Eingehend). Wenn die Sicherheitsgruppe für Ihre Instance korrekt konfiguriert ist, sollte Ihnen eine Regel mit den folgenden Werten angezeigt werden:

   - Typ: HTTP
   - Protocol (Protokoll): TCP
   - Portbereich: 80
   - Quelle: 0.0.0. 0/0

5. Wenn Sie keine Regel mit diesen Werten sehen, verwenden Sie die Verfahren unter Regeln zu einer Sicherheitsgruppe hinzufügen, um sie einer neuen Sicherheitsregel hinzuzufügen.

Schritt 2: Konfigurieren Sie Ihren Quellinhalt für die Bereitstellung auf der Windows Server Amazon EC2 EC2-Instance

Jetzt ist es an der Zeit, den Quellinhalt Ihrer Anwendung so zu konfigurieren, dass Sie etwas haben, das Sie auf der Amazon EC2 EC2-Instance bereitstellen können. In diesem Tutorial stellen Sie eine einzelne Webseite für die Amazon EC2 EC2-Instance bereit, auf der Windows Server ausgeführt wird, auf der Internet Information Services (IIS) als Webserver ausgeführt wird. Auf dieser Webseite wird ein einfaches „Hello, World!“ angezeigt message.

Themen

- Erstellen Sie die Webseite
- Erstellen Sie ein Skript, um Ihre Anwendung auszuführen
- Fügen Sie eine Anwendungsspezifikationsdatei hinzu

Erstellen Sie die Webseite

1. Erstellen Sie ein Unterverzeichnis (Unterordner) mit dem Namen `HelloWorldApp` in Ihrem Ordner `c:\temp` und wechseln Sie dann zu diesem Ordner.

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

> **Note**
>
> Sie müssen nicht den Standort `c:\temp` oder den Unterordnernamen `HelloWorldApp` verwenden. Wenn Sie einen anderen Standort oder einen anderen Unterordnernamen verwenden, müssen Sie diesen durchgehend im ganzen Tutorial verwenden.

2. Erstellen Sie mit einem Texteditor eine Datei in dem Ordner. Benennen Sie die Datei `index.html`.

```
notepad index.html
```

3. Fügen Sie der Datei den folgendes HTML-Code hinzu und speichern Sie dann die Datei.

```
<!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>
```

Erstellen Sie ein Skript, um Ihre Anwendung auszuführen

Als Nächstes erstellen Sie ein Skript, mit dem der Webserver auf der Amazon EC2 EC2-Zielinstanz eingerichtet CodeDeploy wird.

1. Erstellen Sie mithilfe eines Texteditors eine weitere Datei im selben Unterordner, in dem die Datei `index.html` gespeichert ist. Benennen Sie die Datei `before-install.bat`.

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

2. Fügen Sie den folgenden Stapelskriptcode zu der Datei hinzu und speichern Sie dann die Datei.

   ```
   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
   ```

Fügen Sie eine Anwendungsspezifikationsdatei hinzu

Als Nächstes fügen Sie zusätzlich zur Webseite und zur AppSpec Batch-Skriptdatei eine Anwendungsspezifikationsdatei (Datei) hinzu. Die AppSpec Datei ist eine YAML-formatierte Datei, die verwendet wird, um: CodeDeploy

- Ordnen Sie die Quelldateien in Ihrer Anwendungsrevision den entsprechenden Zielen auf der Instance zu.
- Angeben von Skripts, die während der Bereitstellung in der Instance ausgeführt werden.

Die AppSpec Datei muss benannt werden. `appspec.yml` Sie muss sich im Stammordner des Anwendungsquellcodes befinden.

1. Erstellen Sie im selben Unterordner, in dem die Dateien `index.html` und `before-install.bat` gespeichert sind, mit einem Texteditor eine weitere Datei. Benennen Sie die Datei `appspec.yml`.

   ```
   notepad appspec.yml
   ```

2. Fügen Sie der Datei den folgendes YAML-Code hinzu und speichern Sie dann die Datei.

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

CodeDeploy verwendet diese AppSpec Datei, um die `index.html` Datei im Stammordner des Anwendungsquellcodes in den `c:\inetpub\wwwroot` Ordner auf der Amazon EC2 EC2-Zielinstanz zu kopieren. Führt während der Bereitstellung das `before-install.bat` Batch-Skript auf der Amazon EC2 EC2-Zielinstanz während des **`BeforeInstall`** Bereitstellungslebenszyklus aus. CodeDeploy Wenn die Ausführung dieses Skripts länger als 900 Sekunden (15 Minuten) dauert, CodeDeploy wird die Bereitstellung gestoppt und die Bereitstellung auf der Amazon EC2 EC2-Instance als fehlgeschlagen markiert.

Weitere Informationen zu diesen Einstellungen finden Sie in der CodeDeploy AppSpec Dateiverweis.

> **Important**
>
> Die Platzierung und Anzahl von Leerzeichen zwischen den einzelnen Elementen in dieser Datei ist wichtig. Wenn der Abstand falsch ist, CodeDeploy wird ein Fehler ausgelöst, der möglicherweise schwer zu debuggen ist. Weitere Informationen finden Sie unter AppSpec Abstand zwischen den Dateien.

Schritt 3: Lade dein „Hallo, Welt!“ hoch Anwendung auf Amazon S3

Jetzt bereiten Sie Ihre Quellinhalte vor und laden sie an einen Ort hoch, von dem aus Sie sie bereitstellen CodeDeploy können. Die folgenden Anweisungen zeigen Ihnen, wie Sie einen Amazon S3 S3-Bucket bereitstellen, die Dateien der Anwendungsrevision für den Bucket vorbereiten, die Dateien der Revision bündeln und dann die Revision in den Bucket übertragen.

> **Note**
>
> Obwohl dies in diesem Tutorial nicht behandelt wird, können Sie es verwenden, CodeDeploy um Anwendungen von GitHub Repositorys auf Instances bereitzustellen. Weitere Informationen finden Sie unter Integrieren CodeDeploy mit GitHub.

Themen

- Bereitstellen eines Amazon S3 S3-Buckets
- Bereiten Sie die Dateien der Anwendung für den Bucket vor
- Bündeln Sie die Dateien der Anwendung in einer einzigen Archivdatei und übertragen Sie die Archivdatei per Push

Bereitstellen eines Amazon S3 S3-Buckets

Erstellen Sie einen Speichercontainer oder Bucket in Amazon S3 — oder verwenden Sie einen vorhandenen Bucket. Stellen Sie sicher, dass Sie die Revision in den Bucket hochladen können und dass Amazon EC2 EC2-Instances, die in Bereitstellungen verwendet werden, die Revision aus dem Bucket herunterladen können.

Sie können die AWS CLI Amazon S3-Konsole oder die Amazon S3-APIs verwenden, um einen Amazon S3-Bucket zu erstellen. Sobald Sie den Bucket erstellt haben, müssen Sie ihm und Ihrem CodeDeploy-Benutzer Zugriffsberechtigungen erteilen.

> **Note**
>
> Bucket-Namen müssen in Amazon S3 für alle AWS Konten eindeutig sein. Falls Sie **`amzn-s3-demo-bucket`** nicht verwenden können: Versuchen Sie es mit einem anderen Bucket-Namen wie **`amzn-s3-demo-bucket`**, gefolgt von einem Bindestrich und Ihren Initialen oder einem anderen eindeutigen Bezeichner. Anschließend müssen Sie überall im Tutorial **`amzn-s3-demo-bucket`** durch Ihren Bucket-Namen ersetzen.
> Der Amazon S3 S3-Bucket muss in derselben AWS Region erstellt werden, in der Ihre Amazon EC2 EC2-Ziel-Instances gestartet werden. Wenn Sie den Bucket beispielsweise in der Region USA Ost (Nord-Virginia) erstellen, müssen Ihre Amazon EC2 EC2-Ziel-Instances in der Region USA Ost (Nord-Virginia) gestartet werden.

Themen

- So erstellen Sie einen Amazon S3 S3-Bucket (CLI)
- So erstellen Sie einen Amazon S3 S3-Bucket (Konsole)
- Erteilen Sie Berechtigungen für den Amazon S3 S3-Bucket und Ihre AWS Konto

So erstellen Sie einen Amazon S3 S3-Bucket (CLI)

Rufen Sie den mb Befehl auf, um einen Amazon S3 S3-Bucket mit dem Namen zu erstellen**amzn-s3-demo-bucket**:

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

So erstellen Sie einen Amazon S3 S3-Bucket (Konsole)

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Amazon S3 S3-Konsole Create Bucket aus.
3. Geben Sie im Feld Bucket name (Bucket-Name) einen Namen für den Bucket ein.
4. Wählen Sie in der Liste Region die Zielregion aus und wählen Sie dann Create aus.

Erteilen Sie Berechtigungen für den Amazon S3 S3-Bucket und Ihre AWS Konto

Sie müssen über Berechtigungen zum Hochladen in den Amazon S3 S3-Bucket verfügen. Sie können diese Berechtigungen über eine Amazon S3 S3-Bucket-Richtlinie angeben. In der folgenden Amazon S3 S3-Bucket-Richtlinie ermöglicht die Verwendung des Platzhalterzeichens (*) dem AWS Konto `111122223333` beispielsweise das Hochladen von Dateien in ein beliebiges Verzeichnis im Amazon S3 S3-Bucket mit dem Namen`amzn-s3-demo-bucket`:

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

```
                ]
            }
        }
    ]
}
```

Informationen zur Anzeige Ihrer AWS Konto-ID finden Sie unter Ihre AWS Konto-ID finden.

Jetzt ist ein guter Zeitpunkt, um zu überprüfen, ob der Amazon S3 S3-Bucket Download-Anfragen von jeder teilnehmenden Amazon EC2 EC2-Instance zulässt. Sie können dies über eine Amazon S3 S3-Bucket-Richtlinie angeben. In der folgenden Amazon S3 S3-Bucket-Richtlinie ermöglicht die Verwendung des Platzhalterzeichens (*) beispielsweise jeder Amazon EC2 EC2-Instance mit einem angehängten IAM-Instance-Profil, das den ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` enthält, Dateien aus einem beliebigen Verzeichnis im Amazon S3 S3-Bucket mit dem Namen herunterzuladen: `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"
                ]
            }
        }
    ]
}
```

Informationen zum Generieren und Anhängen einer Amazon S3 S3-Bucket-Richtlinie finden Sie unter Beispiele für Bucket-Richtlinien.

Der CodeDeploy Administratorbenutzer, in dem Sie erstellt haben, Schritt 1: Einrichtung muss auch über die Berechtigung verfügen, die Revision in den Amazon S3 S3-Bucket hochzuladen. Eine Möglichkeit, dies zu spezifizieren, ist eine IAM-Richtlinie, die Sie dem Berechtigungssatz des Benutzers hinzufügen, oder einer IAM-Rolle (die Sie dem Benutzer zuweisen). Die folgende IAM-

Richtlinie ermöglicht es dem Benutzer, Revisionen an einer beliebigen Stelle im Amazon S3 S3-Bucket mit dem Namen hochzuladen: `amzn-s3-demo-bucket`

JSON

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

Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch. Informationen zum Hinzufügen einer Richtlinie zu einem Berechtigungssatz finden Sie unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center Benutzerhandbuch.

Bereiten Sie die Dateien der Anwendung für den Bucket vor

Stellen Sie sicher, dass die Webseite, die AppSpec Datei und das Skript auf Ihrem Entwicklungscomputer wie folgt organisiert sind:

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

Bündeln Sie die Dateien der Anwendung in einer einzigen Archivdatei und übertragen Sie die Archivdatei per Push

Bündeln Sie die Dateien in einer Archivdatei (auch Anwendungsrevision genannt).

> **Note**
>
> Möglicherweise werden Ihnen für die Speicherung von Objekten in einem Bucket und die Übertragung von Anwendungsrevisionen in einen oder aus einem Bucket Gebühren berechnet. Weitere Informationen finden Sie unter Amazon S3 – Preise.

1. Wechseln Sie auf dem Entwicklungscomputer in den Ordner, in dem die Dateien abgelegt sind:

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

   > **Note**
   >
   > Wenn Sie nicht in diesen Ordner wechseln, startet die Dateibündelung im jeweils aktuellen Ordner. Beispiel: Wenn der aktuelle Ordner `c:\temp` statt `c:\temp\HelloWorldApp` ist, beginnt der Bündelvorgang mit den Dateien und Unterordnern im Ordner `c:\temp`, in dem u. U. mehr enthalten ist als nur der `HelloWorldApp`-Unterordner.

2. Rufen Sie den create-application Befehl auf, um eine neue Anwendung **`HelloWorld_App`** mit dem folgenden Namen zu registrieren CodeDeploy:

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

3. Rufen Sie den CodeDeploy Push-Befehl auf, um die Dateien zu bündeln, die Revisionen auf Amazon S3 hochzuladen und Informationen CodeDeploy über die hochgeladene Version zu registrieren — alles in einer Aktion.

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

   Dieser Befehl bündelt die Dateien aus dem aktuellen Verzeichnis (mit Ausnahme aller versteckten Dateien) in einer einzigen Archivdatei mit dem Namen`HelloWorld_App.zip`, lädt die Revision in den **`amzn-s3-demo-bucket`** Bucket hoch und registriert Informationen CodeDeploy über die hochgeladene Revision.

Schritt 4: Stellen Sie Ihre Hello World-Anwendung bereit

Jetzt stellen Sie die Version der Hello World-Beispielanwendung bereit, die Sie auf Amazon S3 hochgeladen haben. Sie verwenden die AWS CLI oder die CodeDeploy Konsole, um die Version bereitzustellen und den Fortschritt der Bereitstellung zu überwachen. Im Anschluss an die Bereitstellung der Anwendungsrevision überprüfen Sie die Ergebnisse.

Themen

- Stellen Sie die Revision Ihrer Anwendung bereit mit CodeDeploy
- Überwachen Sie Ihre Bereitstellung und beheben Sie Fehler
- Überprüfen Sie Ihre Bereitstellung

Stellen Sie die Revision Ihrer Anwendung bereit mit CodeDeploy

Sie können Ihre Anwendung mit der CLI oder der Konsole bereitstellen.

Themen

- Mit der CLI stellen Sie die Anwendungsrevision wie folgt bereit:
- Mit der Konsole stellen Sie die Anwendungsrevision wie folgt bereit:

Mit der CLI stellen Sie die Anwendungsrevision wie folgt bereit:

1. Zuerst ist für die Bereitstellung eine Bereitstellungsgruppe erforderlich. Bevor Sie die Bereitstellungsgruppe erstellen können, benötigen Sie allerdings den ARN einer Service-Rolle. Eine Servicerolle ist eine IAM-Rolle, die einem Dienst die Erlaubnis erteilt, in Ihrem Namen zu handeln. In diesem Fall erteilt die Service-Rolle die CodeDeploy Erlaubnis, auf Ihre Amazon EC2 EC2-Instances zuzugreifen, um deren Amazon EC2 EC2-Instance-Tags zu erweitern (zu lesen).

   Sie sollten bereits anhand der Anweisungen unter Erstellen Sie eine Servicerolle (CLI) eine Service-Rolle erstellt haben. Wie Sie den ARN der Service-Rolle abrufen können, erfahren Sie unter Rufen Sie die Servicerolle ARN (CLI) ab.
2. Nachdem Sie den ARN haben, rufen Sie den create-deployment-group Befehl auf, um eine Bereitstellungsgruppe mit dem Namen zu erstellen**`HelloWorld_DepGroup`**, die der genannten **`HelloWorld_App`** Anwendung zugeordnet ist. Verwenden Sie dazu das Amazon EC2 EC2-Instance-Tag named **`CodeDeployDemo`** und die Bereitstellungskonfiguration mit dem **`CodeDeployDefault.OneAtATime`** Namen der Service-Rolle 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**
>
> Der Befehl create-deployment-group bietet Unterstützung für die Erstellung von Triggern, die zum Senden von Amazon SNS SNS-Benachrichtigungen an Themenabonnenten über bestimmte Ereignisse in Bereitstellungen und Instances führen. Der Befehl unterstützt auch Optionen für das automatische Rollback von Bereitstellungen und die Einrichtung von Alarmen, um Bereitstellungen zu beenden, wenn die Überwachungsschwellenwerte in CloudWatch Amazon-Alarmen erreicht werden. Befehle für diese Aktionen sind in diesem Tutorial nicht enthalten.

3. Bevor Sie eine Bereitstellung erstellen, muss der CodeDeploy Agent auf den Instances in Ihrer Bereitstellungsgruppe installiert sein. Sie können den Agenten über die Befehlszeile mit folgendem Befehl mit AWS Systems Manager installieren:

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

   Dieser Befehl erstellt eine Verknüpfung in Systems Manager State Manager, die den CodeDeploy Agenten installiert und dann versucht, ihn jeden Sonntagmorgen um 2:00 Uhr zu aktualisieren. Weitere Informationen zum CodeDeploy Agenten finden Sie unter Mit dem CodeDeploy Agenten arbeiten. Weitere Informationen zu Systems Manager finden Sie unter Was ist AWS Systems Manager.

4. Rufen Sie nun den Befehl create-deployment zum Erstellen einer Bereitstellung auf, die der Anwendung **`HelloWorld_App`**, der Bereitstellungskonfiguration **`CodeDeployDefault.OneAtATime`** und der Bereitstellungsgruppe **`HelloWorld_DepGroup`** zugeordnet ist, unter Verwendung der Anwendungrevision **`HelloWorld_App.zip`** im Bucket **`amzn-s3-demo-bucket`**:

```
aws deploy create-deployment --application-name HelloWorld_App --deployment-config-
name CodeDeployDefault.OneAtATime --deployment-group-name HelloWorld_DepGroup --s3-
location bucket=amzn-s3-demo-bucket,bundleType=zip,key=HelloWorld_App.zip
```

Mit der Konsole stellen Sie die Anwendungsrevision wie folgt bereit:

1. Bevor Sie die CodeDeploy Konsole zum Bereitstellen Ihrer Anwendungsrevision verwenden, benötigen Sie einen ARN für die Servicerolle. Eine Servicerolle ist eine IAM-Rolle, die einem Dienst die Erlaubnis erteilt, in Ihrem Namen zu handeln. In diesem Fall erteilt die Service-Rolle die CodeDeploy Erlaubnis, auf Ihre Amazon EC2 EC2-Instances zuzugreifen, um deren Amazon EC2 EC2-Instance-Tags zu erweitern (zu lesen).

   Sie sollten bereits anhand der Anweisungen unter Erstellen Sie eine Servicerolle (Konsole) eine Service-Rolle erstellt haben. Wie Sie den ARN der Service-Rolle abrufen können, erfahren Sie unter Rufen Sie die Servicerolle ARN ab (Konsole).
2. Jetzt, wo Sie den ARN haben, können Sie die CodeDeploy Konsole verwenden, um Ihre Anwendungsrevision bereitzustellen.

   Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.
3. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
4. Wählen Sie HelloWorld_App.
5. Wählen Sie auf der Registerkarte Deployment groups (Bereitstellungsgruppen), die Option Create deployment group (Bereitstellungsgruppe erstellen) aus.
6. Geben Sie unter Deployment group name (Name der Bereitstellungsgruppe) **`HelloWorld_DepGroup`** ein.
7. Wählen Sie unter Service Role (Servicerolle) den Namen der Servicerolle aus.
8. Wählen Sie unter Bereitstellungstyp die Option aus In-place.

9. Wählen Sie unter Umgebungskonfiguration Amazon EC2 EC2-Instances aus.
10. Behalten Sie unter Agentenkonfiguration mit AWS Systems Manager die Standardeinstellungen bei.
11. Geben Sie für Key (Schlüssel) **`Name`** ein.
12. Geben Sie unter Value (Wert) **`CodeDeployDemo`** ein.
13. Wählen Sie unter Deployment configuration (Bereitstellungskonfiguration) die Option CodeDeployDefault.OneAtATime aus.
14. Deaktivieren Sie unter Load Balancer die Option Enable load balancing (Load Balancing aktivieren).
15. Wählen Si Create deployment group (Bereitstellungsgruppe erstellen).
16. Wählen Sie Create deployment.
17. Wählen Sie in der Bereitstellungsgruppe die Option _ HelloWorld DepGroup
18. Wählen Sie unter Revisionstyp die Option Meine Anwendung ist in Amazon S3 gespeichert und geben Sie dann unter Revisionsort den Speicherort der Version der Hello World-Beispielanwendung ein, die Sie zuvor auf Amazon S3 hochgeladen haben. Ermitteln Sie den Speicherort wie folgt:

    a. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
    b. Wählen Sie in der Liste der Buckets amzn-s3-demo-bucket aus (oder den Namen des Buckets, in den Sie Ihre Anwendungsrevision hochgeladen haben).
    c. Wählen Sie in der Objektliste _ aus. HelloWorld App.zip
    d. Klicken Sie auf der Registerkarte Overview (Übersicht) auf Copy Path (Pfad kopieren).
    e. Kehren Sie zur CodeDeploy Konsole zurück, und fügen Sie in Revision Location den Feldwert Link ein.
19. Wählen Sie für Revision file type (Revisionsdateityp) die Option .zip aus.
20. (Optional) Geben Sie unter Deployment description (Bereitstellungsbeschreibung) einen Kommentar ein.
21. Wählen Sie Create deployment. Auf der Seite Deployments werden nun Informationen zu der neu erstellten Bereitstellung angezeigt.

Überwachen Sie Ihre Bereitstellung und beheben Sie Fehler

Verwenden Sie die Konsole AWS CLI oder die Konsole, um Ihre Bereitstellung zu überwachen und Fehler zu beheben.

Themen

- Über die CLI können Sie die Bereitstellung wie folgt überwachen und Probleme beheben:
- Über die Konsole können Sie die Bereitstellung wie folgt überwachen und Probleme beheben:

Über die CLI können Sie die Bereitstellung wie folgt überwachen und Probleme beheben:

1. Fordern Sie die ID der Bereitstellung an, indem Sie den Befehl list-deployments für die Anwendung **`HelloWorld_App`** und die Bereitstellungsgruppe **`HelloWorld_DepGroup`** aufrufen:

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

2. Rufen Sie den Befehl get-deployment mit der Bereitstellungs-ID auf:

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

3. Der Befehl gibt den allgemeinen Status der Bereitstellung zurück. Wurde die Bereitstellung erfolgreich abgeschlossen, lautet der Wert `Succeeded`.

   Wenn der Gesamtstatus lautet`Failed`, können Sie Befehle wie list-deployment-instances und get-deployment-instance aufrufen, um Fehler zu beheben. Weitere Optionen für die Problembehebung finden Sie unter Analysieren von Protokolldateien zur Untersuchung von Bereitstellungsfehlern auf Instances.

Über die Konsole können Sie die Bereitstellung wie folgt überwachen und Probleme beheben:

Auf der Seite „Bereitstellungen“ in der CodeDeploy Konsole können Sie den Status Ihrer Bereitstellung in der Spalte Status überwachen.

Weitere Informationen zu Ihrer Bereitstellung können Sie wie nachfolgend beschrieben abrufen. Dies ist insbesondere nützlich, falls in der Spalte Status ein anderer Wert als Succeeded steht:

1. Wählen Sie in der Tabelle Deployments (Bereitstellungen) die ID Ihrer Bereitstellung aus. Falls eine Bereitstellung fehlschlägt, wird auf der Detailseite der Bereitstellung eine Meldung mit dem Grund des Fehlschlagens angezeigt.

2. . Weitere Informationen zu den Instanzen der Bereitstellung werden angezeigt. Wenn eine Bereitstellung fehlschlägt, können Sie möglicherweise feststellen, auf welchen Amazon EC2 EC2-Instances und in welchem Schritt die Bereitstellung fehlgeschlagen ist.
3. Wenn Sie weitere Fehler beheben möchten, können Sie eine Technik wie View Instance Details verwenden. Sie können auch die Bereitstellungsprotokolldateien auf einer Amazon EC2 EC2-Instance analysieren. Weitere Informationen finden Sie unter Analysieren von Protokolldateien zur Untersuchung von Bereitstellungsfehlern auf Instances.

Überprüfen Sie Ihre Bereitstellung

Überprüfen Sie nach der erfolgreichen Bereitstellung, ob die Installation funktioniert. Verwenden Sie die öffentliche DNS-Adresse der Amazon EC2 EC2-Instance, um die Webseite in einem Webbrowser anzuzeigen. (Um den öffentlichen DNS-Wert abzurufen, wählen Sie in der Amazon EC2 EC2-Konsole die Amazon EC2 EC2-Instance aus und suchen Sie auf der Registerkarte Beschreibung nach dem Wert in Public DNS.)

Wenn die öffentliche DNS-Adresse Ihrer Amazon EC2 EC2-Instance beispielsweise lautete**`ec2-01-234-567-890.compute-1.amazonaws.com`**, würden Sie die folgende URL verwenden:

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

Wenn der Vorgang erfolgreich war, sollten Sie die Webseite „Hello World!“ sehen.

Schritt 5: Aktualisieren Sie Ihr „Hallo, Welt!“ und stellen Sie es erneut bereit Anwendung

Nachdem Sie Ihre Anwendungsrevision erfolgreich bereitgestellt haben, aktualisieren Sie den Code der Webseite auf dem Entwicklungscomputer und verwenden Sie ihn dann, CodeDeploy um die Website erneut bereitzustellen. Nach der erneuten Bereitstellung sollten Sie die Änderungen an der Amazon EC2 EC2-Instance sehen können.

Themen

- Ändern Sie die Webseite
- Stellen Sie die Site erneut bereit

Ändern Sie die Webseite

1. Gehen Sie zu Ihrem Unterordner `c:\temp\HelloWorldApp`, und modifizieren Sie die Datei `index.html` mit einem Texteditor:

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

2. Ändern Sie den Inhalt der `index.html`-Datei auf die gewünschte Hintergrundfarbe und ändern Sie Text auf der Webseite. Speichern Sie dann die Datei:

```
<!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>
```

Stellen Sie die Site erneut bereit

Nachdem Sie den Code geändert haben, verwenden Sie Amazon S3 und CodeDeploy stellen Sie die Webseite erneut bereit.

Bündeln Sie die Änderungen und laden Sie sie auf Amazon S3 hoch, wie unter beschriebenBündeln Sie die Dateien der Anwendung in einer einzigen Archivdatei und übertragen Sie die Archivdatei per Push. (Wenn Sie diese Anweisungen befolgen, müssen Sie keine neue Anwendung erstellen.) Geben Sie der Revision denselben Schlüssel wie zuvor (**`HelloWorld_App.zip`**). Laden Sie es in denselben Amazon S3 S3-Bucket hoch, den Sie zuvor erstellt haben (z. B.**`amzn-s3-demo-bucket`**).

Verwenden Sie die AWS CLI oder die CodeDeploy Konsole, um die Site erneut bereitzustellen.

Themen

- Erneutes Bereitstellen der Website (CLI)
- Erneutes Bereitstellen der Website (Konsole)

Erneutes Bereitstellen der Website (CLI)

Rufen Sie den Befehl create-deployment zum Erstellen einer Bereitstellung basierend auf der hochgeladenen Revision erneut auf. Verwenden Sie dazu erneut die Anwendung **`HelloWorld_App`**, die Bereitstellungskonfiguration **`CodeDeployDefault.OneAtATime`**, die Bereitstellungsgruppe **`HelloWorld_DepGroup`** und die Revision **`HelloWorld_App.zip`** im Bucket **`amzn-s3-demo-bucket`**:

```
 aws deploy create-deployment --application-name HelloWorld_App --deployment-config-
name CodeDeployDefault.OneAtATime --deployment-group-name HelloWorld_DepGroup --s3-
location bucket=amzn-s3-demo-bucket,bundleType=zip,key=HelloWorld_App.zip
```

Sie können den Status der neuen Bereitstellung überprüfen, wie in Überwachen Sie Ihre Bereitstellung und beheben Sie Fehler beschrieben.

Wenn die Site erneut bereitgestellt CodeDeploy wurde, rufen Sie die Site erneut in Ihrem Webbrowser auf, um zu überprüfen, ob die Hintergrundfarbe und der Text auf der Webseite geändert wurden. (Sie müssen möglicherweise Ihren Browser aktualisieren.) Wenn die Hintergrundfarbe und der Text geändert wurden – herzlichen Glückwunsch! Sie haben Ihre Website geändert und erneut bereitgestellt.

Erneutes Bereitstellen der Website (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter. https://console.aws.amazon.com/codedeploy

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Wählen Sie im Navigationsbereich Applications (Anwendungen) aus.
3. Wählen Sie in der Anwendungsliste die Option HelloWorld_App aus.
4. Klicken Sie auf der Registerkarte Deployments (Bereitstellungen) auf Create deployment (Bereitstellung erstellen).

   a. Wählen Sie in der Liste der Bereitstellungsgruppen die Option HelloWorld_ DepGroup aus.

   b. Geben Sie unter Revisionsort den Amazon S3 S3-Link für Ihre Revision ein.

      Ermitteln des Link-Wertes:

      i. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

         Suchen und öffnen Sie amzn-s3-demo-bucket und wählen Sie dann in der Amazon S3 S3-Konsole Ihre Version aus. **`HelloWorld_App.zip`**

      ii. Wenn der Eigenschaftenbereich in der Amazon S3 S3-Konsole nicht sichtbar ist, wählen Sie die Schaltfläche Eigenschaften.

      iii. Kopieren Sie im Bereich Properties (Eigenschaften) den Wert des Felds Link.

      iv. Kehren Sie zur CodeDeploy Konsole zurück und fügen Sie den Link in den Ordner Revision ein.

      v.

   c. Wenn unter Revision file type (Revisionsdateityp) eine Meldung darauf hinweist, dass der Dateityp nicht erkannt werden konnte, klicken Sie auf .zip.

   d. Lassen Sie das Feld Deployment description (Bereitstellungsbeschreibung) leer.

   e. Erweitern Sie Deployment group overrides (Überschreibungen der Bereitstellungsgruppe). Wählen Sie in der Liste Deployment configuration (Bereitstellungskonfiguration) die Option CodeDeployDefault.OneAtATime aus und klicken Sie dann auf Create deployment (Bereitstellung erstellen).

      Sie können den Status der Bereitstellung überprüfen, wie in Überwachen Sie Ihre Bereitstellung und beheben Sie Fehler beschrieben.

Wenn die Site erneut bereitgestellt CodeDeploy wurde, rufen Sie die Site erneut in Ihrem Webbrowser auf, um zu überprüfen, ob die Hintergrundfarbe und der Text auf der Webseite geändert wurden. (Sie müssen möglicherweise Ihren Browser aktualisieren.) Wenn die Hintergrundfarbe und der Text geändert wurden – herzlichen Glückwunsch! Sie haben Ihre Website geändert und erneut bereitgestellt.

Schritt 6: Räumen Sie Ihr „Hallo, Welt!“ auf Anwendung und zugehörige Ressourcen

Sie haben jetzt erfolgreich ein Update für „Hello, World!“ vorgenommen die Site kodiert und erneut bereitgestellt. Um laufende Gebühren für die Ressourcen, die Sie zum Abschließen dieses Tutorial erstellt haben, zu vermeiden, sollten Sie Folgendes löschen:

- Alle CloudFormation Stacks (oder beenden Sie alle Amazon EC2 EC2-Instances, falls Sie sie außerhalb von CloudFormation erstellt haben).
- Alle Amazon S3 S3-Buckets.
- Die Anwendung `HelloWorld_App` in CodeDeploy.
- Die AWS Systems Manager State Manager-Zuordnung für den CodeDeploy Agenten.

Sie können Amazon S3 AWS CLI CloudFormation, Amazon EC2 und CodeDeploy Konsolen oder die AWS APIs verwenden, um die Bereinigung durchzuführen.

Themen

- So bereinigen Sie Ressourcen (CLI)
- So bereinigen Sie Ressourcen (Konsole):
- Als nächstes

So bereinigen Sie Ressourcen (CLI)

1. Wenn Sie den CloudFormation Stack für dieses Tutorial verwendet haben, löschen Sie den Stack, indem Sie den delete-stack Befehl für den genannten Stack aufrufen. **`CodeDeployDemoStack`** Dadurch werden alle zugehörigen Amazon EC2 EC2-Instances beendet und alle zugehörigen IAM-Rollen gelöscht, die ursprünglich vom Stack erstellt wurden.

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

2. Um den Amazon S3 S3-Bucket zu löschen, rufen Sie den rm Befehl mit dem --recursive Schalter für den genannten Bucket auf**`amzn-s3-demo-bucket`**. Damit werden der Bucket und alle Objekte im Bucket gelöscht.

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

3. Rufen Sie den delete-application Befehl auf CodeDeploy, um die `HelloWorld_App` Anwendung von zu löschen. Dadurch werden alle zugehörigen Bereitstellungsgruppen-Datensätze und die Bereitstellungsdatensätze für die Anwendung gelöscht.

```
aws deploy delete-application --application-name HelloWorld_App
```

4. Rufen Sie den delete-association Befehl auf, um die Systems Manager State Manager-Zuordnung zu löschen.

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

   Sie können den abrufen, *`association-id`* indem Sie den describe-association Befehl aufrufen.

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

5. Wenn Sie den CloudFormation Stack für dieses Tutorial nicht verwendet haben, rufen Sie den terminate-instances Befehl zum Beenden von Amazon EC2 EC2-Instances auf, die Sie manuell erstellt haben. Geben Sie die ID der Amazon EC2 EC2-Instance an, die beendet werden soll.

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

So bereinigen Sie Ressourcen (Konsole):

Wenn Sie unsere CloudFormation Vorlage für dieses Tutorial verwendet haben, löschen Sie den zugehörigen CloudFormation Stack.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

2. Geben Sie in das Suchfeld den Namen des CloudFormation Stacks ein (z. B.**`CodeDeployDemoStack`**).
3. Aktivieren Sie das Kontrollkästchen neben dem Stack-Namen.
4. Wählen Sie im Menü Actions die Option Delete Stack. Dadurch wird der Stack gelöscht, alle zugehörigen Amazon EC2 EC2-Instances beendet und alle zugehörigen IAM-Rollen gelöscht.

So beenden Sie Amazon EC2 EC2-Instances, die Sie außerhalb eines CloudFormation Stacks erstellt haben:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Bereich Instances die Option Instances.
3. Geben Sie in das Suchfeld den Namen der Amazon EC2 EC2-Instance ein, die Sie beenden möchten, und drücken Sie dann die EINGABETASTE.
4. Wählen Sie die Amazon EC2 EC2-Instance aus.
5. Wählen Sie Actions, zeigen Sie auf Instance State, und wählen Sie dann Terminate. Wählen Sie bei Aufforderung Yes, Terminate. Wiederholen Sie diese Schritte für alle weiteren Amazon EC2 EC2-Instances.

Um den Amazon S3 S3-Bucket zu löschen:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Suchen Sie in der Liste der Buckets nach dem Namen des Amazon S3 S3-Buckets und wählen Sie ihn aus (z. B.**`amzn-s3-demo-bucket`**).
3. Bevor Sie einen Bucket löschen können, müssen Sie zunächst seinen Inhalt löschen. Wählen Sie alle Dateien im Bucket aus, z. B. **`HelloWorld_App.zip`**. Wählen Sie im Menü Actions die Option Delete. Wenn Sie zum Bestätigen des Löschvorgangs aufgefordert werden, wählen Sie OK.
4. Nachdem der Bucket leer ist, kann er gelöscht werden. Wählen Sie in der Bucket-Liste die Zeile mit dem Bucket aus (aber nicht den Namen des Buckets). Wählen Sie Delete bucket und bei der Aufforderung zum Bestätigen OK.

Um die `HelloWorld_App` Anwendung zu löschen von CodeDeploy:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.

3. Wählen Sie **`HelloWorld_App`**.
4. Klicken Sie auf Delete Application (Anwendung löschen).
5. Geben Sie bei der Aufforderung **`Delete`** ein und klicken Sie dann auf Delete (Löschen).

So löschen Sie die Systems Manager State Manager-Zuordnung:

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager.
2. Wählen Sie im Navigationsbereich Statusmanager aus.
3. Wählen Sie die Verknüpfung aus, die Sie erstellt haben, und wählen Sie Delete (Löschen) aus.

Als nächstes

Wenn Sie hier angekommen sind, haben Sie eine Bereitstellung mit erfolgreich abgeschlossen CodeDeploy. Herzlichen Glückwunsch!

Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat Enterprise Linux)

Dieses Tutorial hilft Ihnen dabei, Erfahrungen zu sammeln, CodeDeploy indem es Sie durch die Bereitstellung einer Beispielanwendungsrevision auf einer einzelnen lokalen Instanz — d. h. einem physischen Gerät, das keine Amazon EC2 EC2-Instance ist — führt, auf der Windows Server, Ubuntu

Server oder Red Hat Enterprise Linux (RHEL) ausgeführt wird. Informationen zu lokalen Instances und deren Verwendung finden Sie unter. CodeDeploy Working with On-Premises Instances

Nicht das, wonach Sie gesucht haben?

- Informationen zum Bereitstellen auf einer Amazon EC2 EC2-Instance, auf der Amazon Linux oder RHEL ausgeführt wird, finden Sie unter. Tutorial: Bereitstellung WordPress auf einer Amazon EC2 EC2-Instance (Amazon Linux oder Red Hat Enterprise Linux und Linux, macOS oder Unix)
- Informationen zum Bereitstellen auf einer Amazon EC2 EC2-Instance, auf der Windows Server ausgeführt wird, finden Sie unterTutorial: Stellen Sie ein „Hallo, Welt!“ bereit Anwendung mit CodeDeploy (Windows Server).

Themen

- Voraussetzungen
- Schritt 1: Konfigurieren Sie die lokale Instanz
- Schritt 2: Erstellen Sie eine Version der Beispielanwendung
- Schritt 3: Bündeln Sie Ihre Anwendungsrevision und laden Sie sie auf Amazon S3 hoch
- Schritt 4: Stellen Sie die Revision Ihrer Anwendung bereit
- Schritt 5: Überprüfen Sie Ihre Bereitstellung
- Schritt 6: Bereinigen von Ressourcen

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die Voraussetzungen unter erfüllen. Dazu gehören die Konfiguration eines Benutzers, die Installation oder Aktualisierung von und die AWS CLI Erstellung einer Servicerolle. Erste Schritte mit CodeDeploy Sie müssen kein IAM-Instanzprofil erstellen, wie in den Voraussetzungen beschrieben. On-premises Instanzen verwenden keine IAM-Instanzprofile.

Das physische Gerät, das Sie als lokale Instance konfigurieren, muss eines der Betriebssysteme ausführen, die aufgeführt sind in Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy.

Schritt 1: Konfigurieren Sie die lokale Instanz

Bevor Sie etwas auf einer lokalen Instance bereitstellen können, müssen Sie diese konfigurieren. Befolgen Sie die Anweisungen unter Working with On-Premises Instances, und kehren Sie dann zu dieser Seite zurück.

Installieren Sie den Agenten CodeDeploy

Nachdem Sie Ihre lokale Instanz konfiguriert haben, folgen Sie den Schritten für lokale Instanzen unter Den CodeDeploy Agenten installieren und kehren Sie zu dieser Seite zurück.

Schritt 2: Erstellen Sie eine Version der Beispielanwendung

In diesem Schritt erstellen Sie eine Beispielanwendungsrevision, die auf der lokalen Instance bereitgestellt werden soll.

Da es schwierig ist zu wissen, welche Software und Funktionen auf Ihrer lokalen Instanz bereits installiert sind — oder gemäß den Richtlinien Ihres Unternehmens installiert werden dürfen —, verwendet die hier angebotene Version der Beispielanwendung einfach Batchskripts (für Windows Server) oder Shellskripts (für Ubuntu Server und RHEL), um Textdateien an einen Speicherort auf Ihrer lokalen Instanz zu schreiben. Eine Datei wird für jedes von mehreren Ereignissen im CodeDeploy Bereitstellungslebenszyklus geschrieben, einschließlich Install AfterInstall, ApplicationStart, und ValidateService. Während des BeforeInstallBereitstellungslebenszyklus wird ein Skript ausgeführt, um alte Dateien zu entfernen, die während früherer Bereitstellungen dieses Beispiels geschrieben wurden, und um einen Speicherort auf Ihrer lokalen Instanz zu erstellen, in den die neuen Dateien geschrieben werden.

> **Note**
>
> Die Bereitstellung dieser Beispielanwendungsrevision schlägt fehl, wenn eine der folgenden Aussagen wahr ist:
>
> - Der Benutzer, der den CodeDeploy Agenten auf der lokalen Instanz startet, ist nicht berechtigt, Skripts auszuführen.
> - Der Benutzer ist nicht berechtigt, Ordner an den in den Skripten aufgeführten Speicherorten zu erstellen oder zu löschen.
> - Der Benutzer ist nicht berechtigt, Textdateien an den in den Skripten aufgeführten Speicherorten zu erstellen.

> **Note**
>
> Wenn Sie eine Windows Server-Instanz konfiguriert haben und ein anderes Beispiel bereitstellen möchten, sollten Sie das Beispiel aus Schritt 2: Konfigurieren Sie Ihren Quellinhalt für die Bereitstellung auf der Windows Server Amazon EC2 EC2-Instance dem Tutorial: Stellen Sie ein „Hallo, Welt!“ bereit Anwendung mit CodeDeploy (Windows Server) Tutorial verwenden.
> Wenn Sie eine RHEL-Instanz konfiguriert haben und ein anderes Beispiel bereitstellen möchten, sollten Sie das Beispiel aus Schritt 2: Konfigurieren Sie Ihren Quellinhalt für die Bereitstellung auf der Amazon Linux- oder Red Hat Enterprise Linux Amazon EC2 EC2-Instance dem Tutorial: Bereitstellung WordPress auf einer Amazon EC2 EC2-Instance (Amazon Linux oder Red Hat Enterprise Linux und Linux, macOS oder Unix) Tutorial verwenden.
> Derzeit gibt es kein alternatives Beispiel für Ubuntu Server.

1. Erstellen Sie auf dem Bereitstellungscomputer ein Unterverzeichnis (Unterordner) mit der Bezeichnung `CodeDeployDemo-OnPrem`, in dem die Dateien der Beispielanwendungsrevision gespeichert werden, und wechseln Sie zu diesem Unterordner. In diesem Beispiel gehen wir davon aus, dass Sie den `c:\temp` Ordner als Stammordner für Windows Server oder den `/tmp` Ordner als Stammordner für Ubuntu Server und RHEL verwenden. Wenn Sie einen anderen Ordner verwenden, müssen Sie im gesamten Tutorial unseren Ordnernamen durch Ihren Ordnernamen ersetzen:

   Für Windows:

   ```
   mkdir c:\temp\CodeDeployDemo-OnPrem
   cd c:\temp\CodeDeployDemo-OnPrem
   ```

   Für Linux, macOS oder Unix:

   ```
   mkdir /tmp/CodeDeployDemo-OnPrem
   cd /tmp/CodeDeployDemo-OnPrem
   ```

2. Verwenden Sie im Stammverzeichnis des Unterordners `CodeDeployDemo-OnPrem` einen Texteditor zur Erstellung zweier Dateien mit den Bezeichnungen `appspec.yml` und `install.txt`:

`appspec.yml`für 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`für Ubuntu Server und 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
```

Weitere Informationen zu AppSpec Dateien finden Sie unter Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy undCodeDeploy AppSpec Dateiverweis.

`install.txt`:

```
The Install deployment lifecycle event successfully completed.
```

3. Erstellen Sie im Stammverzeichnis des Unterordners `CodeDeployDemo-OnPrem` einen Unterordner `scripts`, und wechseln Sie dann zu diesem Ordner:

   Für Windows:

   ```
   mkdir c:\temp\CodeDeployDemo-OnPrem\scripts
   cd c:\temp\CodeDeployDemo-OnPrem\scripts
   ```

   Für Linux, macOS oder Unix:

   ```
   mkdir -p /tmp/CodeDeployDemo-OnPrem/scripts
   cd /tmp/CodeDeployDemo-OnPrem/scripts
   ```

4. Verwenden Sie einen Texteditor, um im Stammverzeichnis des `scripts` Unterordners vier Dateien mit den Namen `before-install.batafter-install.bat`,`application-start.bat`, und `validate-service.bat` für Windows Server oder, `before-install.sh` `after-install.shapplication-start.sh`, und `validate-service.sh` für Ubuntu Server und RHEL zu erstellen:

   Für 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
```

Für Ubuntu Server und 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. Stellen Sie nur für Ubuntu Server und RHEL sicher, dass die vier Shell-Skripte über Ausführungsberechtigungen verfügen:

```
chmod +x ./scripts/*
```

Schritt 3: Bündeln Sie Ihre Anwendungsrevision und laden Sie sie auf Amazon S3 hoch

Bevor Sie Ihre Anwendungsrevision bereitstellen können, müssen Sie die Dateien bündeln und dann das Dateipaket in einen Amazon S3 S3-Bucket hochladen. Befolgen Sie die Anweisungen unter Erstellen Sie eine Anwendung mit CodeDeploy und Eine Revision CodeDeploy auf Amazon S3 übertragen (nur EC2/On-Premises Bereitstellungen). (Sie können die Anwendung und die Bereitstellungsgruppe zwar beliebig benennen, empfohlen werden jedoch der Anwendungsname `CodeDeploy-OnPrem-App` und der Bereitstellungsgruppenname `CodeDeploy-OnPrem-DG`.) Nachdem Sie diese Anweisungen ausgeführt haben, kehren Sie zu dieser Seite zurück.

> **Note**
>
> Alternativ können Sie das Dateipaket in ein GitHub Repository hochladen und von dort aus bereitstellen. Weitere Informationen finden Sie unter Integrieren CodeDeploy mit GitHub.

Schritt 4: Stellen Sie die Revision Ihrer Anwendung bereit

Nachdem Sie Ihre Anwendungsversion in einen Amazon S3 S3-Bucket hochgeladen haben, versuchen Sie, sie auf Ihrer lokalen Instance bereitzustellen. Befolgen Sie die Anweisungen unter Erstellen Sie eine Bereitstellung mit CodeDeploy, und kehren Sie dann zu dieser Seite zurück.

Schritt 5: Überprüfen Sie Ihre Bereitstellung

Um zu überprüfen, ob die Bereitstellung erfolgreich war, befolgen Sie die Anweisungen unter CodeDeploy Bereitstellungsdetails anzeigen, und dann kehren Sie zu dieser Seite zurück.

Wenn die Bereitstellung erfolgreich war, finden Sie vier Textdateien im `c:\temp\CodeDeployExample` Ordner (für Windows Server) oder `/tmp/CodeDeployExample` (für Ubuntu Server und RHEL).

Schlug die Bereitstellung fehl, dann befolgen Sie Fehlerbehandlungsschritte unter View Instance Details und Beheben von Instance-Problemen. Nehmen Sie die notwendigen Korrekturen vor, packen Sie die Anwendungsrevision erneut, laden Sie sie hoch, und versuchen Sie erneut, die Bereitstellung durchzuführen.

Schritt 6: Bereinigen von Ressourcen

Um laufende Gebühren für Ressourcen zu vermeiden, die Sie für dieses Tutorial erstellt haben, löschen Sie den Amazon S3 S3-Bucket, wenn Sie ihn nicht mehr verwenden werden. Sie können auch zugehörige Ressourcen bereinigen, z. B. die Anwendungs- und Bereitstellungsgruppeneinträge in CodeDeploy der lokalen Instance.

Sie können die AWS CLI oder eine Kombination aus den CodeDeploy und Amazon S3 S3-Konsolen und den verwenden AWS CLI , um Ressourcen zu bereinigen.

Ressourcen bereinigen (CLI)

Um den Amazon S3 S3-Bucket zu löschen

- Rufen Sie den Befehl rm mit der Befehlszeilenoption `--recursive` für den Bucket auf (z. B. `amzn-s3-demo-bucket`). Der Bucket und alle im Bucket enthaltenen Objekte werden gelöscht.

```
aws s3 rm s3://your-bucket-name --recursive --region region
```

Um die Anwendungs- und Bereitstellungsgruppendatensätze in zu löschen CodeDeploy

- Rufen Sie den Befehl delete-application für die Anwendung auf (z. B.). `CodeDeploy-OnPrem-App` Die Datensätze für die Bereitstellung und die Bereitstellungsgruppe werden gelöscht.

```
aws deploy delete-application --application-name your-application-name
```

Um die Registrierung der lokalen Instanz aufzuheben und den IAM-Benutzer zu löschen

- Rufen Sie den Befehl deregister für die lokale Instanz und Region auf:

```
aws deploy deregister --instance-name your-instance-name --delete-iam-user --
region your-region
```

> **Note**
>
> Wenn Sie den mit dieser lokalen Instanz verknüpften IAM-Benutzer nicht löschen möchten, verwenden Sie stattdessen die Option. `--no-delete-iam-user`

Um den CodeDeploy Agenten zu deinstallieren und die Konfigurationsdatei aus der lokalen Instanz zu entfernen

- Rufen Sie von der lokalen Instanz aus den Deinstallationsbefehl auf:

```
aws deploy uninstall
```

Sie haben jetzt alle Schritte ausgeführt, um die Ressourcen zu bereinigen, die in diesem Tutorial verwendet wurden.

Ressourcen bereinigen (Konsole)

Um den Amazon S3 S3-Bucket zu löschen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie das Symbol neben dem Bucket aus, den Sie löschen möchten (z. B. `amzn-s3-demo-bucket`), ohne jedoch den Bucket auszuwählen.
3. Wählen Sie Aktionen und anschließend Löschen aus.
4. Wenn Sie zum Löschen des Buckets aufgefordert werden, wählen Sie OK.

Um die Anwendungs- und Bereitstellungsgruppeneinträge zu löschen CodeDeploy

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Wählen Sie im Navigationsbereich Applications (Anwendungen) aus.
3. Wählen Sie den Namen der Anwendung aus, die Sie löschen möchten (z. B. `CodeDeploy-OnPrem-App`) und klicken Sie auf Delete Application (Anwendung löschen).
4. Geben Sie bei Aufforderung den Namen der Anwendung ein, um das Löschen zu bestätigen, und klicken Sie auf Delete (Löschen).

Sie können die AWS CodeDeploy Konsole nicht verwenden, um die lokale Instanz abzumelden oder den Agenten zu deinstallieren. CodeDeploy Folgen Sie den Anweisungen in Um die Registrierung der lokalen Instanz aufzuheben und den IAM-Benutzer zu löschen.

Tutorial: Wird verwendet CodeDeploy , um eine Anwendung für eine Auto Scaling Scaling-Gruppe bereitzustellen

In diesem Tutorial verwenden Sie, CodeDeploy um eine Anwendungsrevision für eine Auto Scaling Scaling-Gruppe bereitzustellen. Amazon EC2 Auto Scaling startet Amazon EC2 EC2-Instances unter Verwendung vordefinierter Bedingungen und beendet diese Instances dann, wenn sie nicht mehr benötigt werden. Amazon EC2 Auto Scaling kann bei der Skalierung helfen CodeDeploy , indem sichergestellt wird, dass immer die richtige Anzahl von Amazon EC2 EC2-Instances zur Verfügung steht, um die Last für Bereitstellungen zu bewältigen. Informationen zur Amazon EC2 Auto Scaling Scaling-Integration mit finden Sie CodeDeploy unterIntegration CodeDeploy mit Amazon EC2 Auto Scaling.

Themen

- Voraussetzungen
- Schritt 1: Auto Scaling Scaling-Gruppe erstellen und konfigurieren
- Schritt 2: Bereitstellen der Anwendung in der Auto Scaling Scaling-Gruppe
- Schritt 3: Überprüfen Sie Ihre Ergebnisse
- Schritt 4: Erhöhen Sie die Anzahl der Amazon EC2 EC2-Instances in der Auto Scaling Scaling-Gruppe
- Schritt 5: Überprüfen Sie Ihre Ergebnisse erneut
- Schritt 6: Bereinigen

Voraussetzungen

So folgen Sie diesem Tutorial:

- Führen Sie alle Schritte unter ausErste Schritte mit CodeDeploy, einschließlich der Einrichtung und Konfiguration des AWS CLI IAM-Instanzprofils (**`CodeDeployDemo-EC2-Instance-Profile`**) und einer Servicerolle (**`CodeDeployDemo`**). Eine Servicerolle ist eine besondere Art von IAM-Rolle, die einem Dienst die Erlaubnis erteilt, in Ihrem Namen zu handeln.
- Wenn Sie Ihre Auto Scaling Scaling-Gruppe mit einer Startvorlage erstellen, müssen Sie die folgenden Berechtigungen hinzufügen:
  - `ec2:RunInstances`
  - `ec2:CreateTags`

- `iam:PassRole`

  Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle Erstellen einer Startvorlage für eine Auto Scaling Scaling-Gruppe und Unterstützung von Startvorlagen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- Erstellen und verwenden Sie eine Version, die mit einer Ubuntu-Server-Instance kompatibel ist und CodeDeploy. Für Ihre Revision können Sie einen der folgenden Schritte ausführen:
  - Erstellen und verwenden Sie die Beispielrevision in Schritt 2: Erstellen Sie eine Version der Beispielanwendung im Tutorial Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat Enterprise Linux).
  - Informationen zum Erstellen einer eigenen Revision finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.
- Erstellen Sie eine Sicherheitsgruppe **`CodeDeployDemo-AS-SG`** mit dem Namen der folgenden Regel für eingehende Nachrichten:
  - Typ: HTTP
  - Quelle: Irgendwo

  Dies ist erforderlich, um Ihre Anwendung anzuzeigen und den Erfolg der Bereitstellung zu überprüfen. Informationen zum Erstellen einer Sicherheitsgruppe finden Sie unter Erstellen einer Sicherheitsgruppe im Amazon EC2 EC2-Benutzerhandbuch.

Schritt 1: Auto Scaling Scaling-Gruppe erstellen und konfigurieren

In diesem Schritt erstellen Sie eine Auto Scaling Scaling-Gruppe, die eine einzelne Amazon Linux-, RHEL- oder Windows Server Amazon EC2 EC2-Instance enthält. In einem späteren Schritt werden Sie Amazon EC2 Auto Scaling anweisen, eine weitere Amazon EC2-Instance hinzuzufügen, und CodeDeploy Ihre Revision darauf bereitstellen.

Themen

- So erstellen und konfigurieren Sie die Auto Scaling Scaling-Gruppe (CLI)
- So erstellen und konfigurieren Sie die Auto Scaling Scaling-Gruppe (Konsole)

So erstellen und konfigurieren Sie die Auto Scaling Scaling-Gruppe (CLI)

1. Rufen Sie den create-launch-template Befehl auf, um eine Amazon EC2 EC2-Startvorlage zu erstellen.

   Bevor Sie diesen Befehl aufrufen, benötigen Sie die ID eines AMIs, das für dieses Tutorial funktioniert, dargestellt durch den Platzhalter*image-id*. Sie benötigen außerdem den Namen eines Amazon EC2 EC2-Instance-Schlüsselpaars, um den Zugriff auf die Amazon EC2 EC2-Instance zu ermöglichen, dargestellt durch den Platzhalter. *key-name*

   So rufen Sie die ID eines AMI ab, das mit diesem Tutorial funktioniert:

   a. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/.

   b. Wählen Sie im Navigationsbereich unter Instances Instances und dann Launch Instance.

   c. Notieren Sie sich auf der Registerkarte Schnellstart der Seite „Amazon Machine Image auswählen“ die ID des AMI neben Amazon Linux 2 AMI, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS oder Microsoft Windows Server 2012 R2.

   > **Note**
   >
   > Wenn Sie über eine benutzerdefinierte Version eines AMI verfügen, das mit CodeDeploy kompatibel ist, wählen Sie es hier aus, statt über die Registerkarte Quick Start (Schnellstart) zu navigieren. Informationen zur Verwendung eines benutzerdefinierten AMI mit CodeDeploy Amazon EC2 Auto Scaling finden Sie unterVerwenden eines benutzerdefinierten AMI mit CodeDeploy Amazon EC2 Auto Scaling.

   Verwenden Sie für das Amazon EC2 EC2-Instance-Schlüsselpaar den Namen Ihres Amazon EC2 EC2-Instance-Schlüsselpaars.

   Führen Sie den Befehl create-launch-template aus.

   Auf lokalen Linux-, macOS- oder Unix-Computern:

   ```
   aws ec2 create-launch-template \
     --launch-template-name CodeDeployDemo-AS-Launch-Template \
     --launch-template-data file://config.json
   ```

Der Inhalt der `config.json` Datei:

```
{
  "InstanceType":"t1.micro",
  "ImageId":"image-id",
  "IamInstanceProfile":{
    "Name":"CodeDeployDemo-EC2-Instance-Profile"
  },
  "KeyName":"key-name"
}
```

Auf lokalen Computern mit Windows:

```
aws ec2 create-launch-template --launch-template-name CodeDeployDemo-AS-Launch-
Template --launch-template-data file://config.json
```

Der Inhalt der `config.json` Datei:

```
{
  "InstanceType":"t1.micro",
  "ImageId":"image-id",
  "IamInstanceProfile":{
    "Name":"CodeDeployDemo-EC2-Instance-Profile"
  },
  "KeyName":"key-name"
}
```

Diese Befehle erstellen zusammen mit der `config.json` Datei eine Amazon EC2 EC2-Startvorlage mit dem Namen CodeDeployDemo-AS-Launch-Template Ihrer Auto Scaling Scaling-Gruppe, die in einem nächsten Schritt auf der Grundlage des Amazon EC2 EC2-Instance-Typs t1.micro erstellt wird. Basierend auf Ihrer Eingabe für `ImageId`IamInstanceProfile, und spezifiziert die Startvorlage auch die AMI-IDKeyName, den Namen des Instance-Profils, das der IAM-Rolle zugeordnet ist, das beim Start an Instances übergeben werden soll, und das Amazon EC2 EC2-Schlüsselpaar, das für die Verbindung mit Instances verwendet werden soll.

2. Rufen Sie den create-auto-scaling-group Befehl auf, um eine Auto Scaling Scaling-Gruppe zu erstellen. Sie benötigen den Namen einer der Availability Zones in einer der unter Region

aufgeführten Regionen und den Namen der Endpoints in der Allgemeine AWS-Referenz, dargestellt durch den Platzhalter*availability-zone*.

> **Note**
>
> Um eine Liste der Availability Zones in einer Region einzusehen, rufen Sie an:
>
> ```
> aws ec2 describe-availability-zones --region region-name
> ```
>
> Um beispielsweise eine Liste der Availability Zones in der Region USA West (Oregon) einzusehen, rufen Sie an:
>
> ```
> aws ec2 describe-availability-zones --region us-west-2
> ```
>
> Eine Liste der Identifikatoren für Regionsnamen finden Sie unterBucket-Namen für das Resource Kit nach Regionen.

Auf lokalen Linux-, macOS- oder Unix-Computern:

```
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
```

Auf lokalen Computern mit Windows:

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name
 CodeDeployDemo-AS-Group --launch-template LaunchTemplateName=CodeDeployDemo-
AS-Launch-Template,Version="$Latest" --min-size 1 --max-size 1 --
desired-capacity 1 --availability-zones availability-zone --tags
 Key=Name,Value=CodeDeployDemo,PropagateAtLaunch=true
```

Mit diesen Befehlen wird eine Auto Scaling Scaling-Gruppe erstellt, die auf der benannten Amazon EC2 EC2-Startvorlage **CodeDeployDemo-AS-Group** basiert. **CodeDeployDemo-AS-**

`Launch-Template` Diese Auto Scaling Scaling-Gruppe hat nur eine Amazon EC2 EC2-Instance und sie wird in der angegebenen Availability Zone erstellt. Jede Instanz in dieser Auto Scaling Scaling-Gruppe wird das Tag haben`Name=CodeDeployDemo`. Das Tag wird bei der späteren Installation des CodeDeploy Agenten verwendet.

3. Rufen Sie den Befehl describe-auto-scaling-groups für **`CodeDeployDemo-AS-Group`** auf:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
 CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,
 LifecycleState]" --output text
```

   Fahren Sie erst fort, wenn die zurückgegebenen Werte `Healthy` und `InService` anzeigen.

4. Auf den Instances in Ihrer Auto Scaling Scaling-Gruppe muss der CodeDeploy Agent installiert sein, damit er in CodeDeploy Bereitstellungen verwendet werden kann. Installieren Sie den CodeDeploy Agenten, indem Sie den create-association Befehl AWS Systems Manager mit den Tags aufrufen, die bei der Erstellung der Auto Scaling Scaling-Gruppe hinzugefügt wurden.

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

 --parameters action=Install, name=AWSCodeDeployAgent \
  --schedule-expression "cron(0 2 ? * SUN *)"
```

   Dieser Befehl erstellt eine Verknüpfung in Systems Manager State Manager, die den CodeDeploy Agenten auf allen Instances in der Auto Scaling Scaling-Gruppe installiert und dann versucht, ihn jeden Sonntagmorgen um 2:00 Uhr zu aktualisieren. Weitere Informationen zum CodeDeploy Agenten finden Sie unter Mit dem CodeDeploy Agenten arbeiten. Weitere Informationen zu Systems Manager finden Sie unter Was ist AWS Systems Manager.

So erstellen und konfigurieren Sie die Auto Scaling Scaling-Gruppe (Konsole)

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Vergewissern Sie sich, dass in der globalen Navigationsleiste eine der unter Region aufgeführten Regionen und Endpunkte in der ausgewählt Allgemeine AWS-Referenzsind. Amazon EC2 Auto Scaling Scaling-Ressourcen sind an die von Ihnen angegebene Region gebunden und CodeDeploy werden nur in ausgewählten Regionen unterstützt.
3. Wählen Sie in der Navigationsleiste unter Instances die Option Launch Templates aus.

4. Wählen Sie Startvorlage erstellen.
5. Geben Sie im Dialogfeld Name und Beschreibung der Startvorlage als Namen der Startvorlage Folgendes ein**`CodeDeployDemo-AS-Launch-Template`**. Behalten Sie die Standardwerte für die anderen Felder bei.
6. Klicken Sie im Dialogfeld Amazon Machine Image (AMI) auf das Drop-down-Menü unter AMI und wählen Sie ein AMI aus, das mit diesem Tutorial funktioniert:

   - Wählen Sie auf der Registerkarte Schnellstart der AMI-Dropdown-Liste eine der folgenden Optionen aus: Amazon Linux 2 AMI, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS oder Microsoft Windows Server 2012 R2.

     > **Note**
     >
     > Wenn Sie über eine benutzerdefinierte Version eines AMI verfügen, das mit CodeDeploy kompatibel ist, wählen Sie es hier aus, statt über die Registerkarte Quick Start (Schnellstart) zu navigieren. Informationen zur Verwendung eines benutzerdefinierten AMI mit CodeDeploy Amazon EC2 Auto Scaling finden Sie unterVerwenden eines benutzerdefinierten AMI mit CodeDeploy Amazon EC2 Auto Scaling.

7. Wählen Sie unter Instanztyp die Dropdownliste aus und wählen Sie t1.micro aus. Sie können die Suchleiste verwenden, um es schneller zu finden.
8. Wählen Sie im Dialogfeld key pair (Anmeldung) die Option Bestehendes Schlüsselpaar auswählen aus. Wählen Sie in der Drop-down-Liste key pair auswählen das Amazon EC2 EC2-Instance-Schlüsselpaar aus, das Sie in den vorherigen Schritten erstellt oder verwendet haben.
9. Wählen Sie im Dialogfeld Netzwerkeinstellungen die Option Virtual Public Cloud (VPC) aus.

   Wählen Sie in der Dropdownliste Sicherheitsgruppen die Sicherheitsgruppe aus, die Sie im Abschnitt Voraussetzungen des Tutorials erstellt haben ()**`CodeDeployDemo-AS-SG`**.
10. Erweitern Sie das Dialogfeld Erweiterte Details. Wählen Sie in der Dropdownliste für das IAM-Instanzprofil unter IAM-Instanzprofil die IAM-Rolle aus, die Sie zuvor erstellt haben (**`CodeDeployDemo-EC2-Instance-Profile`**).

    Behalten Sie die restlichen Standardeinstellungen bei.
11. Wählen Sie Startvorlage erstellen.
12. Wählen Sie im Dialogfeld „Nächste Schritte“ die Option Auto Scaling Scaling-Gruppe erstellen aus.

13. Geben Sie auf der Seite Startvorlage oder Konfiguration auswählen den Namen der Auto Scaling Scaling-Gruppe ein**`CodeDeployDemo-AS-Group`**.
14. Im Dialogfeld Startvorlage sollte Ihre Startvorlage (**`CodeDeployDemo-AS-Launch-Template`**) ausgefüllt sein. Falls nicht, wählen Sie sie aus dem Dropdownmenü aus. Behalten Sie die Standardeinstellungen bei und wählen Sie Weiter.
15. Wählen Sie auf der Seite „Instance-Startoptionen auswählen“ im Bereich Netzwerk für VPC die Standard-VPC aus. Wählen Sie dann für Availability Zones und Subnetze ein Standardsubnetz aus. Sie müssen eine VPC erstellen, wenn Sie den Standard nicht wählen können. Weitere Informationen finden Sie unter Erste Schritte mit Amazon VPC.
16. Verwenden Sie im Abschnitt Instance type requirements (Anforderungen an den Instance-Typ) die Standardeinstellung, um diesen Schritt zu vereinfachen. (Setzen Sie die Startvorlage nicht außer Kraft.) In diesem Tutorial starten Sie nur On-Demand Instances mit dem in Ihrer Startvorlage angegebenen Instance-Typ.
17. Wählen Sie Next (Weiter), um zur Seite Configure advanced options (Erweiterte Optionen konfigurieren) zu gelangen.
18. Behalten Sie die Standardwerte bei und wählen Sie Weiter.
19. Behalten Sie auf der Seite Gruppengröße und Skalierungsrichtlinien konfigurieren die Standardwerte für Gruppengröße von 1 bei. Wählen Sie Weiter aus.
20. Überspringen Sie den Schritt zur Konfiguration von Benachrichtigungen und wählen Sie Weiter.
21. Fügen Sie auf der Seite Tags hinzufügen ein Tag hinzu, das bei der späteren Installation des CodeDeploy Agenten verwendet werden soll. Wählen Sie Add tag.

    a. Geben Sie für Key (Schlüssel) **`Name`** ein.

    b. Geben Sie unter Value (Wert) **`CodeDeployDemo`** ein.

    Wählen Sie Weiter aus.
22. Überprüfen Sie Ihre Auto Scaling Scaling-Gruppeninformationen auf der Überprüfungsseite und wählen Sie dann Auto Scaling Scaling-Gruppe erstellen.
23. Wählen Sie in der Navigationsleiste, wenn Auto Scaling Scaling-Gruppen ausgewählt sind**`CodeDeployDemo-AS-Group`**, und wählen Sie dann die Registerkarte Instanzverwaltung aus. Fahren Sie erst fort, wenn der Wert von InServicein der Spalte Lifecycle und der Wert Healthy in der Spalte Health Status angezeigt wird.
24. Installieren Sie den CodeDeploy Agenten, indem Sie den Schritten unter CodeDeploy Agent installieren folgen und die `Name=CodeDeployDemo` Instanz-Tags verwenden.

Schritt 2: Bereitstellen der Anwendung in der Auto Scaling Scaling-Gruppe

In diesem Schritt stellen Sie die Revision für die einzelne Amazon EC2 EC2-Instance in der Auto Scaling Scaling-Gruppe bereit.

Themen

- So erstellen Sie die Bereitstellung (CLI)
- So erstellen Sie die Bereitstellung (Konsole)

So erstellen Sie die Bereitstellung (CLI)

1. Rufen Sie den Befehl create-application zum Erstellen einer Anwendung mit dem Namen **`SimpleDemoApp`** auf:

   ```
   aws deploy create-application --application-name SimpleDemoApp
   ```

2. Sie sollten bereits eine Servicerolle anhand der Anweisungen unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy erstellt haben. Die Servicerolle erteilt die CodeDeploy Erlaubnis, auf Ihre Amazon EC2 EC2-Instances zuzugreifen, um deren Tags zu erweitern (zu lesen). Sie benötigen den ARN der Servicerolle. Zum Abrufen des Servicerollen-ARN befolgen Sie die Anweisungen unter Rufen Sie die Servicerolle ARN (CLI) ab.

3. Nachdem Sie über einen Dienstrollen-ARN verfügen, rufen Sie den create-deployment-group Befehl auf, um eine Bereitstellungsgruppe mit dem Namen**`SimpleDemoDG`**, die der genannten Anwendung zugeordnet ist**`SimpleDemoApp`**, mithilfe der Auto Scaling Scaling-Gruppe mit dem Namen **`CodeDeployDemo-AS-Group`** und der angegebenen Bereitstellungskonfiguration mit dem angegebenen Dienstrollen-ARN zu erstellen. **`CodeDeployDefault.OneAtATime`**

   > **Note**
   >
   > Der Befehl create-deployment-group bietet Unterstützung für die Erstellung von Triggern, die zum Senden von Amazon SNS SNS-Benachrichtigungen an Themenabonnenten über bestimmte Ereignisse in Bereitstellungen und Instances führen. Der Befehl unterstützt auch Optionen für das automatische Rollback von Bereitstellungen und die Einrichtung von Alarmen, um Bereitstellungen zu beenden, wenn die Überwachungsschwellenwerte in CloudWatch Amazon-Alarmen erreicht werden. Befehle für diese Aktionen sind in diesem Tutorial nicht enthalten.

Auf lokalen Linux-, macOS- oder Unix-Computern:

```
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
```

Auf lokalen Computern mit Windows:

```
aws deploy create-deployment-group --application-name SimpleDemoApp --auto-scaling-
groups CodeDeployDemo-AS-Group --deployment-group-name SimpleDemoDG --deployment-
config-name CodeDeployDefault.OneAtATime --service-role-arn service-role-arn
```

4. Rufen Sie den Befehl create-deployment zum Erstellen einer Bereitstellung auf, die der Anwendung **SimpleDemoApp**, der Bereitstellungskonfiguration **CodeDeployDefault.OneAtATime** und der Bereitstellungsgruppe **SimpleDemoDG** unter Verwendung der Revision am angegebenen Speicherort zugeordnet wird.

   Für Amazon Linux- und RHEL Amazon EC2 EC2-Instances, Aufrufe von lokalen Linux-, macOS- oder Unix-Computern

   ```
   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*ist der Name des Amazon S3 S3-Buckets, der die CodeDeploy Resource Kit-Dateien für Ihre Region enthält. Ersetzen Sie beispielsweise für die Region USA Ost (Ohio) *bucket-name* durch`aws-codedeploy-us-east-2`. Eine Liste der Bucket-Namen finden Sie unterBucket-Namen für das Resource Kit nach Regionen.

   Für Amazon Linux- und RHEL Amazon EC2 EC2-Instances, Aufrufen von lokalen Windows-Computern

```
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*ist der Name des Amazon S3 S3-Buckets, der die CodeDeploy Resource Kit-Dateien für Ihre Region enthält. Ersetzen Sie beispielsweise für die Region USA Ost (Ohio) *bucket-name* durch`aws-codedeploy-us-east-2`. Eine Liste der Bucket-Namen finden Sie unterBucket-Namen für das Resource Kit nach Regionen.

Für Windows Server Amazon EC2 EC2-Instances, Aufrufe von lokalen Linux-, macOS- oder Unix-Computern

```
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*ist der Name des Amazon S3 S3-Buckets, der die CodeDeploy Resource Kit-Dateien für Ihre Region enthält. Ersetzen Sie beispielsweise für die Region USA Ost (Ohio) *bucket-name* durch`aws-codedeploy-us-east-2`. Eine Liste der Bucket-Namen finden Sie unterBucket-Namen für das Resource Kit nach Regionen.

Für Windows Server Amazon EC2 EC2-Instances, Aufrufen von lokalen Windows-Computern

```
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*ist der Name des Amazon S3 S3-Buckets, der die CodeDeploy Resource Kit-Dateien für Ihre Region enthält. Ersetzen Sie beispielsweise für die Region USA Ost (Ohio) *bucket-name* durch`aws-codedeploy-us-east-2`. Eine Liste der Bucket-Namen finden Sie unterBucket-Namen für das Resource Kit nach Regionen.

> **Note**
>
> Derzeit CodeDeploy bietet es keine Beispielversion für die Bereitstellung auf Ubuntu Server Amazon EC2 EC2-Instances. Informationen zum eigenen Erstellen einer Revision finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.

5. Rufen Sie den Befehl get-deployment auf, um sicherzustellen, dass die Bereitstellung erfolgreich war.

   Bevor Sie diesen Befehl aufrufen, benötigen Sie die ID der Bereitstellung, die vom Aufruf des Befehls create-deployment zurückgegeben worden sein sollte. Wenn Sie die Bereitstellungs-ID erneut benötigen, rufen Sie den Befehl list-deployments für die Anwendung **`SimpleDemoApp`** und die Bereitstellungsgruppe **`SimpleDemoDG`** auf:

   ```
   aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-
   name SimpleDemoDG --query "deployments" --output text
   ```

   Rufen Sie nun den Befehl get-deployment mit der Bereitstellungs-ID auf:

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

   Fahren Sie erst fort, wenn der zurückgegebene Wert `Succeeded` ist.

So erstellen Sie die Bereitstellung (Konsole)

1. Sie sollten bereits eine Servicerolle anhand der Anweisungen unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy erstellt haben. Die Service-Rolle CodeDeploy erteilt Ihnen die Erlaubnis, auf Ihre Instances zuzugreifen, um deren Tags zu erweitern (zu lesen). Bevor Sie die CodeDeploy Konsole zum Bereitstellen Ihrer Anwendungsrevision verwenden, benötigen Sie die Dienstrolle ARN. Zum Abrufen des Servicerollen-ARN befolgen Sie die Anweisungen unter Rufen Sie die Servicerolle ARN ab (Konsole).
2. Da Sie nun über die Dienstrolle ARN verfügen, können Sie die CodeDeploy Konsole verwenden, um Ihre Anwendungsrevision bereitzustellen.

   Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

3. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
4. Wählen Sie Create application aus.
5. Wählen Sie Custom application (Benutzerdefinierte Anwendung).
6. Geben Sie unter Application name (Anwendungsname) **`SimpleDemoApp`** ein.
7. Wählen Sie unter Rechenplattform die Option EC2/On-premises.
8. Wählen Sie Create application aus.
9. Wählen Sie auf der Registerkarte Deployment groups (Bereitstellungsgruppen), die Option Create deployment group (Bereitstellungsgruppe erstellen) aus.
10. Geben Sie unter Deployment group name (Name der Bereitstellungsgruppe) **`SimpleDemoDG`** ein.
11. Wählen Sie unter Service Role (Servicerolle) den Namen Ihrer Servicerolle aus.
12. Wählen Sie unter Bereitstellungstyp die Option aus In-place.
13. Wählen Sie in der Umgebungskonfiguration Auto Scaling Scaling-Gruppen aus und wählen Sie dann**`CodeDeployDemo-AS-Group`**.
14. Wählen Sie unter Deployment configuration (Bereitstellungskonfiguration) die Option CodeDeployDefault.OneAtATime aus.
15. Deaktivieren Sie Enable load balancing (Load Balancing aktivieren).
16. Wählen Si Create deployment group (Bereitstellungsgruppe erstellen).
17. Klicken Sie auf der Seite der Bereitstellungsgruppe auf Create deployment (Bereitstellung erstellen).
18. Wählen Sie unter Revisionstyp die Option Meine Anwendung ist in Amazon S3 gespeichert aus.
19. Geben Sie im Feld Revisionsort den Speicherort der Beispielanwendung für Ihr Betriebssystem und Ihre Region ein.

    Für Amazon Linux- und RHEL Amazon EC2 EC2-Instances

Region	Speicherort der Beispielanwendung
Region USA Ost (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip`
Region USA Ost (Nord-Virginia)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip`
Region USA West (Nordkalifornien)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip`
Region USA West (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip`
Region Kanada (Zentral)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip`
Region Europa (Irland)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip`
Region Europa (London)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip`

Region	Speicherort der Beispielanwendung
Region Europa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip`
Region Europa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip`
Region Israel (Tel Aviv)	`https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Linux.zip`
Asien-Pazifik (Hongkong)-Region	`https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Linux.zip`
Asia Pacific (Tokyo) Region	`http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip`
Asia Pacific (Seoul) Region	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip`
Region Asien-Pazifik (Singapur)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip`

Region	Speicherort der Beispielanwendung
Region Asien-Pazifik (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip`
Region Asien-Pazifik (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Linux.zip`
Region Asien-Pazifik (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip`
Region Südamerika (São Paulo)	`http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip`

Für Windows Server Amazon EC2 EC2-Instances

Region	Speicherort der Beispielanwendung
Region USA Ost (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip`
Region USA Ost (Nord-Virginia)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip`

Region	Speicherort der Beispielanwendung
Region USA West (Nordkalifornien)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip`
Region USA West (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip`
Region Kanada (Zentral)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip`
Region Europa (Irland)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip`
Region Europa (London)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip`
Region Europa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip`
Region Europa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip`

Region	Speicherort der Beispielanwendung
Region Israel (Tel Aviv)	`https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Region Asien-Pazifik (Hongkong)	`https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Asia Pacific (Seoul) Region	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip`
Region Asien-Pazifik (Singapur)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip`
Region Asien-Pazifik (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip`
Region Asien-Pazifik (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Region Asien-Pazifik (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip`

Region	Speicherort der Beispielanwendung
Region Südamerika (São Paulo)	`http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip`

Für Ubuntu Server Amazon EC2 EC2-Instances

Geben Sie den Speicherort Ihrer benutzerdefinierten Anwendungsversion ein, die in Amazon S3 gespeichert ist.

20. Lassen Sie das Feld Deployment description (Bereitstellungsbeschreibung) leer.
21. Erweitern Sie Advanced (Erweitert).
22. Wählen Sie Create deployment.

> **Note**
>
> Wenn Failed (Fehlgeschlagen) anstelle von Succeeded (Erfolgreich) angezeigt wird, sollten Sie einige der Methoden in Überwachen Sie Ihre Bereitstellung und beheben Sie Fehler (mit dem Anwendungsnamen **`SimpleDemoApp`** und dem Bereitstellungsgruppennamen **`SimpleDemoDG`**) ausprobieren.

Schritt 3: Überprüfen Sie Ihre Ergebnisse

In diesem Schritt überprüfen Sie, ob die **`SimpleDemoApp`** Version auf der einzelnen Amazon EC2 EC2-Instance in der Auto Scaling Scaling-Gruppe CodeDeploy installiert wurde.

Themen

- So prüfen Sie die Ergebnisse (CLI)
- So prüfen Sie die Ergebnisse (Konsole)

So prüfen Sie die Ergebnisse (CLI)

Zunächst benötigen Sie das öffentliche DNS der Amazon EC2 EC2-Instance.

Verwenden Sie die AWS CLI , um das öffentliche DNS der Amazon EC2 EC2-Instance in der Auto Scaling Scaling-Gruppe abzurufen, indem describe-instances Sie den Befehl aufrufen.

Bevor Sie diesen Befehl aufrufen, benötigen Sie die ID der Amazon EC2 EC2-Instance. Zum Anfordern der ID rufen Sie wie bereits zuvor den describe-auto-scaling-groups für **`CodeDeployDemo-AS-Group`** auf:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --output text
```

Rufen Sie nun den Befehl describe-instances auf:

```
aws ec2 describe-instances --instance-id instance-id --query
 "Reservations[0].Instances[0].PublicDnsName" --output text
```

Der zurückgegebene Wert ist das öffentliche DNS der Amazon EC2 EC2-Instance.

Zeigen Sie mit einem Webbrowser die für diese Amazon EC2 EC2-Instance bereitgestellte SimpleDemoApp Version an, und verwenden Sie dabei eine URL wie die folgende:

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

Wenn Sie die Seite mit den Glückwünschen sehen, haben Sie erfolgreich eine Revision CodeDeploy für eine einzelne Amazon EC2 EC2-Instance in einer Auto Scaling Scaling-Gruppe bereitgestellt!

Als Nächstes fügen Sie der Auto Scaling Scaling-Gruppe eine Amazon EC2 EC2-Instance hinzu. Nachdem Amazon EC2 Auto Scaling die Amazon EC2 EC2-Instance hinzugefügt hat, CodeDeploy wird Ihre Revision auf der neuen Instance bereitgestellt.

So prüfen Sie die Ergebnisse (Konsole)

Zunächst benötigen Sie das öffentliche DNS der Amazon EC2 EC2-Instance.

Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/.

Wählen Sie im Amazon EC2 EC2-Navigationsbereich unter Auto Scaling die Option Auto Scaling Groups und dann den **`CodeDeployDemo-AS-Group`** Eintrag aus.

Wählen Sie auf der Registerkarte Instances die Amazon EC2 EC2-Instance-ID in der Liste aus.

Notieren Sie sich auf der Seite Instances auf der Registerkarte Description den Public DNS-Wert. Sie sollte wie folgt aussehen: **`ec2-01-234-567-890.compute-1.amazonaws.com`**.

Zeigen Sie mit einem Webbrowser die für diese Amazon EC2 EC2-Instance bereitgestellte SimpleDemoApp Version an, und verwenden Sie dabei eine URL wie die folgende:

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

Wenn Sie die Seite mit den Glückwünschen sehen, haben Sie erfolgreich eine Revision CodeDeploy für eine einzelne Amazon EC2 EC2-Instance in einer Auto Scaling Scaling-Gruppe bereitgestellt!

Als Nächstes fügen Sie der Auto Scaling Scaling-Gruppe eine Amazon EC2 EC2-Instance hinzu. Nachdem Amazon EC2 Auto Scaling die Amazon EC2 EC2-Instance hinzugefügt hat, CodeDeploy wird Ihre Version auf der neuen Amazon EC2 EC2-Instance bereitgestellt.

Schritt 4: Erhöhen Sie die Anzahl der Amazon EC2 EC2-Instances in der Auto Scaling Scaling-Gruppe

In diesem Schritt weisen Sie die Auto Scaling Scaling-Gruppe an, eine zusätzliche Amazon EC2 EC2-Instance zu erstellen. Nachdem Amazon EC2 Auto Scaling die Instance erstellt hat, CodeDeploy stellt es Ihre Revision darauf bereit.

Themen

- Um die Anzahl der Amazon EC2 EC2-Instances in der Auto Scaling-Gruppe (CLI) zu skalieren
- Um die Anzahl der Amazon EC2 EC2-Instances in der Bereitstellungsgruppe (Konsole) zu skalieren

Um die Anzahl der Amazon EC2 EC2-Instances in der Auto Scaling-Gruppe (CLI) zu skalieren

1. Rufen Sie den update-auto-scaling-group Befehl auf, um die Anzahl der Amazon EC2-Instances in der Auto Scaling Scaling-Gruppe **`CodeDeployDemo-AS-Group`** von eins auf zwei zu erhöhen.

   Auf lokalen Linux-, macOS- oder Unix-Computern:

   ```
   aws autoscaling update-auto-scaling-group \
     --auto-scaling-group-name CodeDeployDemo-AS-Group \
     --min-size 2 \
     --max-size 2 \
     --desired-capacity 2
   ```

Auf lokalen Computern mit Windows:

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-
AS-Group --min-size 2 --max-size 2 --desired-capacity 2
```

2. Stellen Sie sicher, dass die Auto Scaling Scaling-Gruppe jetzt über zwei Amazon EC2 EC2-Instances verfügt. Rufen Sie den Befehl describe-auto-scaling-groups für **`CodeDeployDemo-AS-Group`** auf:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
 CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,
 LifecycleState]" --output text
```

Fahren Sie erst fort, wenn die beiden zurückgegebenen Werte `Healthy` und `InService` anzeigen.

Um die Anzahl der Amazon EC2 EC2-Instances in der Bereitstellungsgruppe (Konsole) zu skalieren

1. Wählen Sie in der Amazon EC2 EC2-Navigationsleiste unter Auto Scaling die Option Auto Scaling Groups und dann. **`CodeDeployDemo-AS-Group`**
2. Wählen Sie Actions und anschließend Bearbeiten.
3. Geben Sie auf der Registerkarte Details in den Feldern Desired (Gewünscht), Min und Max den Wert **2** ein. Klicken Sie dann auf Save (Speichern).
4. Wählen Sie die Registerkarte Instances aus. Die neue Amazon EC2 EC2-Instance sollte in der Liste erscheinen. (Wenn die Instance nicht angezeigt wird, müssen Sie möglicherweise ein paar Mal auf die Schaltfläche Refresh klicken.) Fahren Sie erst fort, wenn der Wert von InServicein der Spalte Lifecycle und der Wert Healthy in der Spalte Health Status angezeigt wird.

Schritt 5: Überprüfen Sie Ihre Ergebnisse erneut

In diesem Schritt überprüfen Sie, ob die SimpleDemoApp Revision auf der neuen Instance in der Auto Scaling Scaling-Gruppe CodeDeploy installiert ist.

Themen

- So prüfen Sie die Ergebnisse der automatischen Bereitstellung (CLI)

- So prüfen Sie die Ergebnisse der automatischen Bereitstellung (Konsole)

So prüfen Sie die Ergebnisse der automatischen Bereitstellung (CLI)

1. Bevor Sie den Befehl get-deployment aufrufen, benötigen Sie die ID der automatischen Bereitstellung. Um die ID abzurufen, rufen Sie den Befehl list-deployments für die Anwendung **`SimpleDemoApp`** und die Bereitstellungsgruppe **`SimpleDemoDG`** auf:

   ```
   aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-
   name SimpleDemoDG --query "deployments" --output text
   ```

   Es sollten zwei Bereitstellungs-IDs vorhanden sein. Verwenden Sie die ID, die Sie noch nicht in einem Aufruf des Befehls get-deployment verwendet haben:

   ```
   aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.
   [status, creator]" --output text
   ```

   Zusätzlich zum Bereitstellungsstatus sollten Sie dies `autoScaling` in der Befehlsausgabe sehen. (`autoScaling`bedeutet, dass Amazon EC2 Auto Scaling die Bereitstellung erstellt hat.)

   Fahren Sie erst fort, wenn der Bereitstellungsstatus `Succeeded` zeigt.

2. Bevor Sie den describe-instances Befehl aufrufen, benötigen Sie die ID der neuen Amazon EC2 EC2-Instance. Zum Abrufen dieser ID rufen Sie erneut den Befehl describe-auto-scaling-groups für **`CodeDeployDemo-AS-Group`** auf:

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
    CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --
   output text
   ```

   Rufen Sie nun den Befehl describe-instances auf:

   ```
   aws ec2 describe-instances --instance-id instance-id --query
    "Reservations[0].Instances[0].PublicDnsName" --output text
   ```

   Notieren Sie sich in der Ausgabe des describe-instances Befehls den öffentlichen DNS für die neue Amazon EC2 EC2-Instance.

3. Zeigen Sie mit einem Webbrowser die für diese Amazon EC2 EC2-Instance bereitgestellte `SimpleDemoApp` Version an, und verwenden Sie dabei eine URL wie die folgende:

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

   Wenn die Seite mit den Glückwünschen angezeigt wird, haben Sie früher eine Revision CodeDeploy für eine skalierte Amazon EC2 EC2-Instance in einer Auto Scaling Scaling-Gruppe bereitgestellt!

So prüfen Sie die Ergebnisse der automatischen Bereitstellung (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich Bereitstellen und wählen Sie dann Bereitstellungen aus.

3. Wählen Sie die Bereitstellungs-ID der Bereitstellung, die Amazon EC2 Auto Scaling erstellt hat.

   .

4. Auf der Seite Deployment (Bereitstellung) werden Informationen über die Bereitstellung angezeigt. Normalerweise würden Sie selbst eine Bereitstellung erstellen, aber Amazon EC2 Auto Scaling hat eine in Ihrem Namen erstellt, um Ihre Version auf der neuen Amazon EC2-Instance bereitzustellen.
5. Sobald als Bereitstellungsstatus oben auf der Seite Succeeded (Erfolgreich) angezeigt wird, überprüfen Sie die Ergebnisse für die Instance. Sie müssen zunächst den öffentlichen DNS der Instance abrufen:
6. Wählen Sie im Amazon EC2 EC2-Navigationsbereich unter Auto Scaling die Option Auto Scaling Groups und dann den **`CodeDeployDemo-AS-Group`** Eintrag aus.
7. Wählen Sie auf der Registerkarte Instances die ID der neuen Amazon EC2 EC2-Instance aus.
8. Notieren Sie sich auf der Seite Instances auf der Registerkarte Description den Public DNS-Wert. Sie sollte wie folgt aussehen: **`ec2-01-234-567-890.compute-1.amazonaws.com`**.

Zeigen Sie die Revision `SimpleDemoApp`, die für die Instanz bereitgestellt wurde, über eine URL wie die folgende an:

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

Wenn die Seite mit den Glückwünschen angezeigt wird, haben Sie früher eine Revision CodeDeploy für eine skalierte Amazon EC2 EC2-Instance in einer Auto Scaling Scaling-Gruppe bereitgestellt!

Schritt 6: Bereinigen

In diesem Schritt löschen Sie die Auto Scaling Scaling-Gruppe, um laufende Gebühren für Ressourcen zu vermeiden, die Sie in diesem Tutorial verwendet haben. Optional können Sie die Auto Scaling Scaling-Konfiguration und die CodeDeploy Deployment-Komponentendatensätze löschen.

Themen

- So bereinigen Sie die Ressourcen (CLI):
- So bereinigen Sie Ressourcen (Konsole):

So bereinigen Sie die Ressourcen (CLI):

1. Löschen Sie die Auto Scaling Scaling-Gruppe, indem delete-auto-scaling-group Sie den Befehl gegen aufrufen**`CodeDeployDemo-AS-Group`**. Dadurch werden auch die Amazon EC2-Instances beendet.

   ```
   aws autoscaling delete-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-AS-Group --force-delete
   ```

2. Löschen Sie optional die Auto Scaling Scaling-Startvorlage, indem Sie den delete-launch-template Befehl für die Startkonfiguration mit dem Namen aufrufen**`CodeDeployDemo-AS-Launch-Template`**:

   ```
   aws ec2 delete-launch-template --launch-template-name CodeDeployDemo-AS-Launch-Template
   ```

3. Löschen Sie optional die Anwendung von, CodeDeploy indem Sie den delete-application Befehl für die angegebene Anwendung aufrufen**`SimpleDemoApp`**. Dadurch werden auch alle zugehörigen Bereitstellungs-, Bereitstellungsgruppen- und Revisions-Datensätze gelöscht.

```
aws deploy delete-application --application-name SimpleDemoApp
```

4. Rufen Sie den delete-association Befehl auf, um die Systems Manager State Manager-Zuordnung zu löschen.

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

   Sie können den abrufen, *association-id* indem Sie den describe-association Befehl aufrufen.

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

So bereinigen Sie Ressourcen (Konsole):

Um die Auto Scaling Scaling-Gruppe zu löschen, die auch die Amazon EC2 EC2-Instances beendet, gehen Sie wie folgt vor:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Amazon EC2 EC2-Navigationsbereich unter Auto Scaling die Option Auto Scaling Groups und dann den **`CodeDeployDemo-AS-Group`** Eintrag aus.
3. Wählen Sie Actions, wählen Sie Delete und dann Yes, Delete.

(Optional) Um die Startvorlage zu löschen:

1. Wählen Sie in der Navigationsleiste unter Auto Scaling die Option Konfigurationen starten und wählen Sie dann**`CodeDeployDemo-AS-Launch-Template`**.
2. Wählen Sie Actions, wählen Sie Delete launch configuration und dann Yes, Delete.

1. Löschen Sie optional die Anwendung von CodeDeploy. Dadurch werden auch alle zugehörigen Bereitstellungs-, Bereitstellungsgruppen- und Revisions-Datensätze gelöscht. Öffnen Sie die CodeDeploy -Konsole unter https://console.aws.amazon.com/codedeploy.
2. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.

3. Wählen Sie aus der Liste der Anwendungen SimpleDemoApp aus.
4. Wählen Sie auf der Seite Application details die Option Delete application.
5. Geben Sie bei der Aufforderung **`Delete`** ein und klicken Sie dann auf Delete (Löschen).

So löschen Sie die Systems Manager State Manager-Zuordnung:

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager.
2. Wählen Sie im Navigationsbereich Statusmanager aus.
3. Wählen Sie die Verknüpfung aus, die Sie erstellt haben, und wählen Sie Delete (Löschen) aus.

Tutorial: Verwenden Sie CodeDeploy zum Bereitstellen einer Anwendung von GitHub

In diesem Tutorial verwenden Sie, CodeDeploy um eine Beispielanwendungsrevision von GitHub für eine einzelne Amazon EC2 EC2-Instance bereitzustellen, auf der Amazon Linux ausgeführt wird, eine einzelne Red Hat Enterprise Linux (RHEL) -Instance oder eine einzelne Windows Server-Instance. Informationen zur GitHub Integration mit CodeDeploy finden Sie unter. Integrieren CodeDeploy mit GitHub

> **Note**
>
> Sie können es auch verwenden CodeDeploy , um eine Anwendungsversion von GitHub auf einer Ubuntu Server-Instanz bereitzustellen. Sie können die unter beschriebene Beispielrevision verwenden oder eine Version erstellenTutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server

oder Red Hat Enterprise Linux), die mit einer Ubuntu-Server-Instanz und kompatibel ist CodeDeploy. Schritt 2: Erstellen Sie eine Version der Beispielanwendung Wie Sie eine eigene Revision erstellen, erfahren Sie unter Planen Sie eine Überarbeitung für CodeDeploy und Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy.

Themen

- Voraussetzungen
- Schritt 1: Richten Sie ein GitHub Konto ein
- Schritt 2: Erstellen Sie ein GitHub Repository
- Schritt 3: Laden Sie eine Beispielanwendung in Ihr GitHub Repository hoch
- Schritt 4: Stellen Sie eine Instanz bereit
- Schritt 5: Erstellen Sie eine Anwendung und eine Bereitstellungsgruppe
- Schritt 6: Stellen Sie die Anwendung auf der Instanz bereit
- Schritt 7: Überwachen und verifizieren Sie die Bereitstellung
- Schritt 8: Bereinigen

Voraussetzungen

Führen Sie die folgenden Schritte durch, bevor Sie mit diesem Tutorial beginnen:

- Installieren Sie Git auf Ihrem lokalen Computer. Informationen zur Installation von Git finden Sie unter Git-Downloads.
- Führen Sie die unter Erste Schritte mit CodeDeploy beschriebenen Schritte durch, einschließlich der Installation und Konfiguration der AWS CLI. Dies ist besonders wichtig, wenn Sie den verwenden möchten AWS CLI , um eine Revision von der Instanz aus GitHub bereitzustellen.

Schritt 1: Richten Sie ein GitHub Konto ein

Sie benötigen ein GitHub Konto, um ein GitHub Repository zu erstellen, in dem die Revision gespeichert wird. Wenn Sie bereits ein GitHub Konto haben, fahren Sie mit fortSchritt 2: Erstellen Sie ein GitHub Repository.

1. Wechseln Sie zu https://github.com/join.

2. Geben Sie einen Benutzernamen, eine E-Mail-Adresse und ein Passwort ein.
3. Wählen Sie Anmelden für GitHub und folgen Sie dann den Anweisungen.

Schritt 2: Erstellen Sie ein GitHub Repository

Sie benötigen ein GitHub Repository, um die Revision zu speichern.

Wenn Sie bereits über ein GitHub Repository verfügen, stellen Sie sicher, dass Sie es **`CodeDeployGitHubDemo`** in diesem Tutorial durch seinen Namen ersetzen, und fahren Sie dann mit fortSchritt 3: Laden Sie eine Beispielanwendung in Ihr GitHub Repository hoch.

1. Führen Sie auf der GitHub Startseite einen der folgenden Schritte aus:
   - Wählen Sie in Your repositories die Option New repository.
   - Wählen Sie auf der Navigationsleiste Create new (+) und dann New repository.
2. Gehen Sie auf der Seite Create a new repository wie folgt vor:
   - Geben Sie in das Textfeld Repository name (Repository-Name) **`CodeDeployGitHubDemo`** ein.
   - Wählen Sie die Option Public aus.

     > **Note**
     >
     > Die Auswahl der Standardoption Public bedeutet, dass alle dieses Repository sehen können. Sie können die Option Private (Privat) auswählen, um zu begrenzen, wer das Repository sehen und ein Commit dafür durchführen kann.

   - Deaktivieren Sie das Kontrollkästchen Initialize this repository with a README. Sie werden stattdessen im nächsten Schritt manuell eine Datei namens `README.md` erstellen.
   - Wählen Sie Repository erstellen aus.
3. Befolgen Sie die für Ihren lokalen Computer passende Anleitung, um das Repository über die Befehlszeile zu erstellen.

   > **Note**
   >
   > Wenn Sie die Zwei-Faktor-Authentifizierung aktiviert haben GitHub, stellen Sie sicher, dass Sie Ihr persönliches Zugriffstoken anstelle Ihres GitHub Anmeldekennworts

eingeben, wenn Sie zur Eingabe eines Kennworts aufgefordert werden. Weitere Informationen finden Sie unter Bereitstellung Ihres 2FA-Authentifizierungscodes.

Auf lokalen Linux-, macOS- oder Unix-Computern:

1. Führen Sie im Terminal nacheinander die folgenden Befehle aus, wobei Ihr GitHub Benutzername angegeben *user-name* ist:

```
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. Lassen Sie unter `/tmp/CodeDeployGitHubDemo` das Terminal offen.

Auf lokalen Computern mit Windows:

1. Rufen Sie eine Eingabeaufforderung als Administrator auf und führen Sie nacheinander die folgenden Befehle aus:

```
mkdir c:\temp\CodeDeployGitHubDemo
```

```
cd c:\temp\CodeDeployGitHubDemo
```

```
notepad README.md
```

2. Speichern Sie die Datei `README.md` im Editor. Schließen Sie den Editor. Führen Sie nacheinander die folgenden Befehle aus, wobei Ihr GitHub Benutzername angegeben *user-name* ist:

```
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. Lassen Sie unter `c:\temp\CodeDeployGitHubDemo` die Befehlszeile offen.

Schritt 3: Laden Sie eine Beispielanwendung in Ihr GitHub Repository hoch

In diesem Schritt kopieren Sie eine Beispielrevision aus einem öffentlichen Amazon S3 S3-Bucket in Ihr GitHub Repository. (Der Einfachheit halber handelt es sich bei den für dieses Tutorial bereitgestellten Beispielrevisionen um einzelne Webseiten.)

> **Note**
>
> Falls Sie statt unserer Beispielrevision eine eigene Revision verwenden möchten, gilt für diese Revision Folgendes:
>
> - Sie muss die unter Planen Sie eine Überarbeitung für CodeDeploy und Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy beschriebenen Richtlinien erfüllen.
> - Sie muss mit dem ausgewählten Instance-Typ kompatibel sein.

- Seien Sie von Ihrem GitHub Dashboard aus zugänglich.

Wenn Ihre Revision diese Anforderungen erfüllt, können Sie direkt mit Schritt 5: Erstellen Sie eine Anwendung und eine Bereitstellungsgruppe fortfahren.
Wenn Sie auf einer Ubuntu Server-Instanz bereitstellen, müssen Sie eine Version, die mit einer Ubuntu-Server-Instanz kompatibel ist, in Ihr GitHub Repository hochladen und CodeDeploy. Weitere Informationen erhalten Sie unter Planen Sie eine Überarbeitung für CodeDeploy und Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy.

Themen

- Eine Beispielrevision von einem lokalen Linux-, macOS- oder Unix-Computer übertragen
- Übertragen einer Beispielrevision per Push von einem lokalen Computer mit Windows

Eine Beispielrevision von einem lokalen Linux-, macOS- oder Unix-Computer übertragen

Führen Sie bei geöffnetem Terminal, beispielsweise unter `/tmp/CodeDeployGitHubDemo` nacheinander die nachfolgenden Befehle aus.

> **Note**
>
> Wenn Sie die Bereitstellung auf einer Windows Server-Instanz planen, ersetzen `SampleApp_Windows.zip` Sie dies `SampleApp_Linux.zip` in den Befehlen.

```
(Amazon S3 copy command)
```

```
unzip SampleApp_Linux.zip
```

```
rm SampleApp_Linux.zip
```

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

Wo *(Amazon S3 copy command)* ist einer der folgenden:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip . --region us-east-2` für die Region USA Ost (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip . --region us-east-1` für die Region USA Ost (Nord-Virginia)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip . --region us-west-1` für die Region USA West (Nordkalifornien)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip . --region us-west-2` für die Region USA West (Oregon)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip . --region ca-central-1`für die Region Kanada (Zentral)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip . --region eu-west-1`für die Region Europa (Irland)
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip . --region eu-west-2`für die Region Europa (London)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip . --region eu-west-3`für die Region Europa (Paris)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip . --region eu-central-1`für die Region Europa (Frankfurt)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Linux.zip . --region il-central-1`für die Region Israel (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Linux.zip . --region ap-east-1`für die Region Asien-Pazifik (Hongkong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip . --region ap-northeast-1`für die Region Asien-Pazifik (Tokio)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip . --region ap-northeast-2`für die Region Asien-Pazifik (Seoul)

- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip . --region ap-southeast-1`für die Region Asien-Pazifik (Singapur)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip . --region ap-southeast-2`für die Region Asien-Pazifik (Sydney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Linux.zip . --region ap-southeast-4`für die Region Asien-Pazifik (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip . --region ap-south-1`für die Region Asien-Pazifik (Mumbai)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip . --region sa-east-1`für die Region Südamerika (São Paulo)

Übertragen einer Beispielrevision per Push von einem lokalen Computer mit Windows

Führen Sie die folgenden Befehle nacheinander aus, während Ihre Eingabeaufforderung z. B. immer noch geöffnet ist: `c:\temp\CodeDeployGitHubDemo`

> **Note**
>
> Wenn Sie die Bereitstellung auf einer Amazon Linux- oder RHEL-Instance planen, ersetzen Sie dies `SampleApp_Windows.zip` in `SampleApp_Linux.zip` den Befehlen.

```
(Amazon S3 copy command)
```

Entpacken Sie den Inhalt der ZIP-Datei `the` direkt in das lokale Verzeichnis (zum Beispiel `c:\temp\CodeDeployGitHubDemo`), nicht in ein neues Unterverzeichnis.

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

Wo *(Amazon S3 copy command)* ist einer der folgenden:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip . --region us-east-2` für die Region USA Ost (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip . --region us-east-1` für die Region USA Ost (Nord-Virginia)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip . --region us-west-1` für die Region USA West (Nordkalifornien)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip . --region us-west-2` für die Region USA West (Oregon)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip . --region ca-central-1`für die Region Kanada (Zentral)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip . --region eu-west-1`für die Region Europa (Irland)
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip . --region eu-west-2`für die Region Europa (London)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip . --region eu-west-3`für die Region Europa (Paris)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip . --region eu-central-1`für die Region Europa (Frankfurt)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Windows.zip . --region il-central-1`für die Region Israel (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Windows.zip . --region ap-east-1`für die Region Asien-Pazifik (Hongkong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Windows.zip . --region ap-northeast-1`für die Region Asien-Pazifik (Tokio)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip . --region ap-northeast-2`für die Region Asien-Pazifik (Seoul)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip . --region ap-southeast-1`für die Region Asien-Pazifik (Singapur)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip . --region ap-southeast-2`für die Region Asien-Pazifik (Sydney)

- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Windows.zip . --region ap-southeast-4`für die Region Asien-Pazifik (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip . --region ap-south-1`für die Region Asien-Pazifik (Mumbai)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip . --region sa-east-1`für die Region Südamerika (São Paulo)

Um Ihre eigene Revision auf eine Ubuntu Server-Instanz zu übertragen, kopieren Sie Ihre Revision in Ihr lokales Repo und rufen Sie dann Folgendes auf:

```
git add .
git commit -m "Added Ubuntu app"
git push
```

Schritt 4: Stellen Sie eine Instanz bereit

In diesem Schritt erstellen oder konfigurieren Sie die Instance, in der Sie die Beispielanwendung bereitstellen werden. Sie können die Bereitstellung auf einer Amazon EC2 EC2-Instance oder einer lokalen Instance durchführen, auf der eines der Betriebssysteme ausgeführt wird, die von unterstützt werden. CodeDeploy Weitere Informationen hierzu erhalten Sie unter Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy. (Wenn Sie bereits eine Instance für die Verwendung in CodeDeploy Bereitstellungen konfiguriert haben, fahren Sie mit dem nächsten Schritt fort.)

Um eine Instanz bereitzustellen

1. Folgen Sie den Anweisungen unter Starten Sie eine Amazon EC2 EC2-Instance (Konsole) So stellen Sie eine Instanz bereit.
2. Denken Sie beim Starten der Instance daran, auf der Seite Tags hinzufügen ein Tag anzugeben. Einzelheiten zur Angabe des Tags finden Sie unterStarten Sie eine Amazon EC2 EC2-Instance (Konsole).

Um zu überprüfen, ob der CodeDeploy Agent auf der Instance ausgeführt wird

- Folgen Sie den Anweisungen unterStellen Sie sicher, dass der CodeDeploy Agent läuft, um zu überprüfen, ob der Agent ausgeführt wird.

Nachdem Sie die Instanz erfolgreich bereitgestellt und überprüft haben, ob der CodeDeploy Agent ausgeführt wird, fahren Sie mit dem nächsten Schritt fort.

Schritt 5: Erstellen Sie eine Anwendung und eine Bereitstellungsgruppe

In diesem Schritt verwenden Sie die CodeDeploy Konsole oder die, AWS CLI um eine Anwendung und eine Bereitstellungsgruppe zu erstellen, mit der Sie die Beispielversion aus Ihrem GitHub Repository bereitstellen können.

Erstellen einer Anwendung und einer Bereitstellungsgruppe (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.

3. Wählen Sie Create application (Anwendung erstellen) und anschließend Custom application (Benutzerdefinierte Anwendung).
4. Geben Sie unter Application name (Anwendungsname) **`CodeDeployGitHubDemo-App`** ein.
5. Wählen Sie unter Compute Platform die Option aus EC2/On-premises.
6. Wählen Sie Create application aus.
7. Wählen Sie auf der Registerkarte Deployment groups (Bereitstellungsgruppen), die Option Create deployment group (Bereitstellungsgruppe erstellen) aus.
8. Geben Sie unter Deployment group name (Name der Bereitstellungsgruppe) **`CodeDeployGitHubDemo-DepGrp`** ein.
9. Wählen Sie unter Servicerolle den Namen Ihrer CodeDeploy Servicerolle aus, die Sie unter Servicerolle erstellen für erstellt haben CodeDeploy.
10. Wählen Sie unter Bereitstellungstyp die Option aus In-place.
11. Wählen Sie in der Umgebungskonfiguration je nach Art der Instance, die Sie verwenden, Amazon EC2 EC2-Instances oder On-premises Instances aus. Geben Sie in den Feldern Key

(Schlüssel) und Value (Wert) den Schlüssel und den Wert des Instance-Tags ein, der auf Ihre Instance als Teil von Schritt 4: Stellen Sie eine Instanz bereit angewendet wurde.

12. Wählen Sie unter Deployment configuration (Bereitstellungskonfiguration) die Option CodeDeployDefault.AllatOnce aus.
13. Deaktivieren Sie unter Load Balancer die Option Enable load balancing (Load Balancing aktivieren).
14. Erweitern Sie Advanced (Erweitert).
15. Wählen Sie unter Alarms (Alarme) die Option Ignore alarm configuration (Alarmkonfiguration ignorieren) aus.
16. Wählen Sie Create deployment group (Bereitstellungsgruppe erstellen) und fahren Sie mit dem nächsten Schritt fort.

Erstellen einer Anwendung und einer Bereitstellungsgruppe (CLI)

1. Rufen Sie den Befehl create-application zum Erstellen einer Anwendung in CodeDeploy mit dem Namen `CodeDeployGitHubDemo-App` auf:

```
aws deploy create-application --application-name CodeDeployGitHubDemo-App
```

2. Rufen Sie den Befehl create-deployment-group auf, um eine Bereitstellungsgruppe mit dem Namen `CodeDeployGitHubDemo-DepGrp` zu erstellen:

   - Wenn Sie auf einer Amazon EC2 EC2-Instance bereitstellen, *`ec2-tag-key`* ist dies der Amazon EC2 EC2-Instance-Tag-Schlüssel, der als Teil von auf Ihre Amazon EC2 EC2-Instance angewendet wurde. Schritt 4: Stellen Sie eine Instanz bereit
   - Wenn Sie auf einer Amazon EC2 EC2-Instance bereitstellen, *`ec2-tag-value`* ist dies der Amazon EC2 EC2-Instance-Tag-Wert, der als Teil von auf Ihre Amazon EC2 EC2-Instance angewendet wurde. Schritt 4: Stellen Sie eine Instanz bereit
   - Wenn Sie die Bereitstellung auf einer lokalen Instance durchführen, *`on-premises-tag-key`* ist dies der lokale Instance-Tag-Schlüssel, der als Teil von auf Ihre lokale Instance angewendet wurde. Schritt 4: Stellen Sie eine Instanz bereit
   - Wenn Sie die Bereitstellung auf einer lokalen Instanz durchführen, *`on-premises-tag-value`* ist dies der Wert des Tags für die lokale Instanz, der als Teil von auf Ihre lokale Instanz angewendet wurde. Schritt 4: Stellen Sie eine Instanz bereit

- *service-role-arn*ist der Dienstrollen-ARN für die Servicerolle, die Sie unter Eine Servicerolle erstellen für erstellt haben CodeDeploy. (Befolgen Sie die Anweisungen unter Rufen Sie die Servicerolle ARN (CLI) ab, um den ARN der erforderlichen Service-Rolle zu ermitteln.)

```
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**
>
> Der Befehl create-deployment-group bietet Unterstützung für die Erstellung von Triggern, die zum Senden von Amazon SNS SNS-Benachrichtigungen an Themenabonnenten über bestimmte Ereignisse in Bereitstellungen und Instances führen. Der Befehl unterstützt auch Optionen für das automatische Rollback von Bereitstellungen und die Einrichtung von Alarmen, um Bereitstellungen zu beenden, wenn die Überwachungsschwellenwerte in CloudWatch Amazon-Alarmen erreicht werden. Befehle für diese Aktionen sind in diesem Tutorial nicht enthalten.

Schritt 6: Stellen Sie die Anwendung auf der Instanz bereit

In diesem Schritt verwenden Sie die CodeDeploy Konsole oder die, AWS CLI um die Beispielrevision aus Ihrem GitHub Repository auf Ihrer Instanz bereitzustellen.

Mit der Konsole stellen Sie die Revision wie folgt bereit:

1. Klicken Sie auf der Seite Deployment group details (Details der Bereitstellungsgruppe) auf Create deployment (Bereitstellung erstellen).
2. Wählen Sie unter Deployment group (Bereitstellungsgruppe) die Option **`CodeDeployGitHubDemo-DepGrp`** aus.
3. Wählen Sie unter Revisionstyp die Option GitHub.
4. Führen Sie unter GitHub Verbinden mit einen der folgenden Schritte aus:

- Um eine Verbindung für CodeDeploy Anwendungen mit einem GitHub Konto herzustellen, melden Sie sich GitHub in einem separaten Webbrowser-Tab ab. Geben Sie GitHub unter Konto einen Namen ein, um diese Verbindung zu identifizieren, und wählen Sie dann Connect GitHub. Auf der Webseite werden Sie aufgefordert, die Interaktion mit GitHub der genannten Anwendung CodeDeploy zu autorisieren. `CodeDeployGitHubDemo-App` Fahren Sie fort mit Schritt 5.
- Um eine Verbindung zu verwenden, die Sie bereits erstellt haben, wählen Sie GitHubunter Konto ihren Namen und dann Connect aus GitHub. Fahren Sie fort mit Schritt 7.
- Um eine Verbindung zu einem anderen GitHub Konto herzustellen, melden Sie sich GitHub in einem separaten Webbrowser-Tab ab. Wählen Sie Connect einem anderen GitHub Konto verbinden und dann Connect GitHub. Fahren Sie fort mit Schritt 5.

5. Folgen Sie den Anweisungen auf der Anmeldeseite, um sich mit Ihrem GitHub Konto anzumelden.
6. Wählen Sie auf der Seite Authorize application die Option Authorize application.
7. Geben Sie auf der Seite Einrichtung CodeDeploy erstellen im Feld Repository-Name den GitHub Benutzernamen ein, mit dem Sie sich angemeldet haben, gefolgt von einem Schrägstrich (/), gefolgt von dem Namen des Repositorys, in das Sie Ihre Anwendungsrevision übertragen haben (z. B.***my-github-user-name*/CodeDeployGitHubDemo**).

   Falls Sie sich nicht sicher sind, welchen Wert Sie eingeben sollen, oder ein anderes Repository angeben möchten:

   a. Rufen Sie in einem separaten Webbrowser-Tab Ihr GitHub Dashboard auf.
   b. Platzieren Sie unter Your repositories den Mauszeiger auf dem Namen des Ziel-Repositorys. Es erscheint ein Tooltip mit dem Namen des GitHub Benutzers oder der Organisation, gefolgt von einem Schrägstrich (/), gefolgt vom Namen des Repositorys. Geben Sie diesen Wert in Repository name (Repository-Name) ein.

   > **Note**
   >
   > Wenn der Name des Ziel-Repositorys nicht in Ihren Repositorys angezeigt wird, verwenden Sie das GitHubSuchfeld, um das Ziel-Repository und den GitHub Benutzer- oder Organisationsnamen zu suchen.

8. Geben Sie in das Feld Commit-ID die ID des Commits ein, der mit der Übertragung der Revision Ihrer Anwendung verknüpft ist. GitHub

Falls Sie sich nicht sicher sind, welchen Wert Sie eingeben müssen:

a. Rufen Sie in einem separaten Webbrowser-Tab Ihr GitHub Dashboard auf.

b. Wählen Sie unter Your repositories (Ihre Repositorys) CodeDeployGitHubDemo aus.

c. Suchen Sie in der Liste der Commits nach der Commit-ID, die dem Push Ihrer Anwendungsrevision zugeordnet ist, und kopieren Sie sie nach GitHub. Diese ID ist in der Regel 40 Zeichen lang und besteht aus Buchstaben und Zahlen. (Verwenden Sie nicht die kürzere Version der Commit-ID. Sie besteht in der Regel aus den ersten zehn Zeichen der längeren Version.)

d. Fügen Sie die Commit-ID in das Feld Commit ID ein.

9. Wählen Sie Deploy, und fahren Sie mit dem nächsten Schritt fort.

Mit der CLI stellen Sie die Revision wie folgt bereit:

Bevor Sie AWS CLI Befehle aufrufen können, mit denen Sie interagieren GitHub (z. B. den create-deployment Befehl, den Sie als Nächstes aufrufen), müssen Sie die CodeDeploy Erlaubnis erteilen, Ihr GitHub Benutzerkonto GitHub für die Interaktion mit der `CodeDeployGitHubDemo-App` Anwendung zu verwenden. Das können Sie derzeit nur über die CodeDeploy -Konsole tun.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.

3. Wählen Sie CodeDeployGitHubDemo-App.

4. Klicken Sie auf der Registerkarte Deployments (Bereitstellungen) auf Create deployment (Bereitstellung erstellen).

> **Note**
>
> Sie erstellen keine neue Bereitstellung. Dies ist derzeit die einzige Möglichkeit, die CodeDeploy Erlaubnis zu erteilen, GitHub im Namen Ihres GitHub Benutzerkontos mit Ihnen zu interagieren.

5. Wählen Sie unter Bereitstellungsgruppe die Option aus CodeDeployGitHubDemo-DepGrp.
6. Wählen Sie unter Revisionstyp die Option aus GitHub.
7. Führen Sie unter GitHub Verbinden mit einen der folgenden Schritte aus:

   - Um eine Verbindung für CodeDeploy Anwendungen mit einem GitHub Konto herzustellen, melden Sie sich GitHub in einem separaten Webbrowser-Tab ab. Geben Sie GitHub unter Konto einen Namen ein, um diese Verbindung zu identifizieren, und wählen Sie dann Connect aus GitHub. Auf der Webseite werden Sie aufgefordert, die Interaktion mit GitHub der genannten Anwendung CodeDeploy zu autorisieren. `CodeDeployGitHubDemo-App` Fahren Sie fort mit Schritt 8.
   - Um eine Verbindung zu verwenden, die Sie bereits erstellt haben, wählen Sie GitHubunter Konto ihren Namen und dann Connect aus GitHub. Fahren Sie fort mit Schritt 10.
   - Um eine Verbindung zu einem anderen GitHub Konto herzustellen, melden Sie sich GitHub in einem separaten Webbrowser-Tab ab. Wählen Sie Connect einem anderen GitHub Konto verbinden und dann Connect GitHub. Fahren Sie fort mit Schritt 8.
8. Folgen Sie den Anweisungen auf der Anmeldeseite, um sich mit Ihrem GitHub Benutzernamen oder Ihrer E-Mail-Adresse und Ihrem Passwort anzumelden.
9. Wählen Sie auf der Seite Authorize application die Option Authorize application.
10. Wählen Sie auf der Seite Bereitstellung CodeDeploy erstellen die Option Abbrechen aus.
11. Rufen Sie den create-deployment Befehl auf, um die Revision aus Ihrem GitHub Repository für die Instance bereitzustellen. Dabei gilt:

    - *`repository`*ist Ihr GitHub Kontoname, gefolgt von einem Schrägstrich (`/`), gefolgt vom Namen Ihres Repositorys (`CodeDeployGitHubDemo`), zum Beispiel. `MyGitHubUserName/CodeDeployGitHubDemo`

      Falls Sie sich nicht sicher sind, welchen Wert Sie verwenden sollen, oder ein anderes Repository angeben möchten:

      1. Gehen Sie in einem separaten Webbrowser-Tab zu Ihrem GitHub Dashboard.

2. Platzieren Sie unter Your repositories den Mauszeiger auf dem Namen des Ziel-Repositorys. Es erscheint ein Tooltip mit dem Namen des GitHub Benutzers oder der Organisation, gefolgt von einem Schrägstrich (/), gefolgt vom Namen des Repositorys. Das ist der Wert, den Sie verwenden müssen.

   > **Note**
   >
   > Wenn der Name des Ziel-Repositorys nicht in Ihren Repositorys erscheint, verwenden Sie das GitHubSuchfeld, um das Ziel-Repository und den entsprechenden GitHub Benutzer- oder Organisationsnamen zu finden.

- *commit-id*ist der Commit, der mit der Version der Anwendungsrevision verknüpft ist, die Sie in Ihr Repository übertragen haben (z. B.`f835159a...528eb76f`).

  Falls Sie sich nicht sicher sind, welchen Wert Sie verwenden müssen:

  1. Rufen Sie in einem separaten Webbrowser-Tab Ihr GitHub Dashboard auf.
  2. Wählen Sie unter Your repositories (Ihre Repositorys) CodeDeployGitHubDemo aus.
  3. Suchen Sie in der Liste der Commits nach der Commit-ID, die dem Push Ihrer Anwendungsrevision zugeordnet GitHub ist. Diese ID ist in der Regel 40 Zeichen lang und besteht aus Buchstaben und Zahlen. (Verwenden Sie nicht die kürzere Version der Commit-ID. Sie besteht in der Regel aus den ersten zehn Zeichen der längeren Version.) Verwenden Sie diesen Wert.

Wenn Sie auf einem lokalen Linux-, macOS- oder Unix-Computer arbeiten:

```
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
```

Wenn Sie auf einem lokalen Computer mit Windows arbeiten:

```
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
```

Schritt 7: Überwachen und verifizieren Sie die Bereitstellung

In diesem Schritt verwenden Sie die CodeDeploy Konsole oder die, AWS CLI um den Erfolg der Bereitstellung zu überprüfen. Außerdem öffnen Sie die in der erstellten oder konfigurierten Instance bereitgestellte Webseite in Ihrem Webbrowser.

> **Note**
>
> Wenn Sie die Bereitstellung auf einer Ubuntu Server-Instanz durchführen, verwenden Sie Ihre eigene Teststrategie, um festzustellen, ob die bereitgestellte Version auf der Instanz erwartungsgemäß funktioniert, und fahren Sie dann mit dem nächsten Schritt fort.

Mit der Konsole überwachen und überprüfen Sie die Bereitstellung wie folgt:

1. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Bereitstellungen aus.

2. Suchen Sie in der Liste der Bereitstellungen nach der Zeile mit dem Wert Application CodeDeployGitHubDemo-Appund dem Wert Deployment Group von. CodeDeployGitHubDemo-DepGrp Wenn weder Succeeded noch Failed in der Spalte Status angezeigt werden, betätigen Sie mehrmals die Schaltfläche Refresh.
3. Wenn Failed in der Spalte Status angezeigt wird, befolgen Sie die Anweisungen in Instanzdetails anzeigen (Konsole), um die Probleme mit der Bereitstellung zu beheben.
4. Wenn Succeeded in der Spalte Status angezeigt wird, können Sie die Bereitstellung in Ihrem Webbrowser überprüfen. Unsere Beispielrevision stellt eine einzelne Webseite in der Instance bereit. Wenn Sie die Bereitstellung auf einer Amazon EC2 EC2-Instance durchführen, gehen Sie in Ihrem Webbrowser zu `http://public-dns` für die Instance (z. B.`http://ec2-01-234-567-890.compute-1.amazonaws.com`).
5. Wenn Sie die Webseite sehen können, sind keine Fehler aufgetreten. Nachdem Sie das Deployment einer Version von erfolgreich verwendet AWS CodeDeploy haben GitHub, können Sie mit dem nächsten Schritt fortfahrenSchritt 8: Bereinigen.

Mit der CLI überwachen und überprüfen Sie die Bereitstellung wie folgt:

1. Rufen Sie den Befehl list-deployments auf, um die Bereitstellungs-ID für die Anwendung `CodeDeployGitHubDemo-App` und die Bereitstellungsgruppe `CodeDeployGitHubDemo-DepGrp` abzurufen:

   ```
   aws deploy list-deployments --application-name CodeDeployGitHubDemo-App --
   deployment-group-name CodeDeployGitHubDemo-DepGrp --query "deployments" --output
    text
   ```

2. Rufen Sie den Befehl get-deployment mit der Bereitstellungs-ID aus der Ausgabe des Befehls list-deployments auf:

   ```
   aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.
   [status, creator]" --output text
   ```

3. Wenn Failed zurückgegeben wird: Befolgen Sie die Anweisungen unter Instanzdetails anzeigen (Konsole), um die Probleme mit der Bereitstellung zu beheben.
4. Wenn Succeeded zurückgegeben wird: Überprüfen Sie die Bereitstellung in Ihrem Webbrowser. Unsere Beispielrevision ist eine einzelne Webseite, die in der Instance bereitgestellt wird. Wenn Sie die Bereitstellung auf einer Amazon EC2 EC2-Instance durchführen, können Sie diese Seite in Ihrem Webbrowser aufrufen, indem Sie zu `http://public-dns` der Amazon EC2 EC2-Instance gehen (z. B.`http://ec2-01-234-567-890.compute-1.amazonaws.com`).
5. Wenn Sie die Webseite sehen können, sind keine Fehler aufgetreten. Sie haben die Bereitstellung aus Ihrem GitHub Repository erfolgreich verwendet AWS CodeDeploy .

Schritt 8: Bereinigen

Um weitere Gebühren für Ressourcen zu vermeiden, die Sie in diesem Tutorial verwendet haben, müssen Sie die Amazon EC2 EC2-Instance und die zugehörigen Ressourcen beenden. Optional können Sie auch die zu diesem Tutorial gehörenden CodeDeploy-Bereitstellungskomponentendatensätze löschen. Wenn Sie ein GitHub Repository nur für dieses Tutorial verwendet haben, können Sie es jetzt auch löschen.

Um ein zu löschen CloudFormation Stapel (wenn du das benutzt hast CloudFormation Vorlage zum Erstellen einer Amazon EC2 EC2-Instanz)

1. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Spalte Stacks den Stack aus, der mit `CodeDeploySampleStack` beginnt.
3. Wählen Sie Löschen aus.
4. Wählen Sie bei Aufforderung Delete stack (Stack löschen) aus. Die Amazon EC2 EC2-Instance und das zugehörige IAM-Instance-Profil und die Service-Rolle werden gelöscht.

Heben Sie wie folgt manuell die Registrierung einer lokalen Instance auf und bereinigen Sie sie (falls Sie eine lokale Instance bereitgestellt haben):

1. Verwenden Sie den AWS CLI , um den Befehl deregister für die lokale Instance aufzurufen, die hier dargestellt wird durch *`your-instance-name`* und die zugehörige Region durch: *`your-region`*

   ```
   aws deploy deregister --instance-name your-instance-name --no-delete-iam-user --
   region your-region
   ```

2. Rufen Sie von der lokalen Instanz aus den Befehl uninstall auf:

   ```
   aws deploy uninstall
   ```

Um eine Amazon EC2 EC2-Instance manuell zu beenden (wenn Sie eine Amazon EC2 EC2-Instance manuell gestartet haben)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances die Option Instances.
3. Wählen Sie das Kästchen neben der Amazon EC2 EC2-Instance aus, die Sie beenden möchten. Zeigen Sie im Menü Actions auf Instance State, und wählen Sie dann Terminate.
4. Wählen Sie bei Aufforderung Yes, Terminate.

Um die Datensätze der CodeDeploy Bereitstellungskomponenten zu löschen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.

3. Wählen Sie CodeDeployGitHubDemo-App.
4. Klicken Sie auf Delete Application (Anwendung löschen).
5. Geben Sie bei der Aufforderung **`Delete`** ein und klicken Sie dann auf Delete (Löschen).

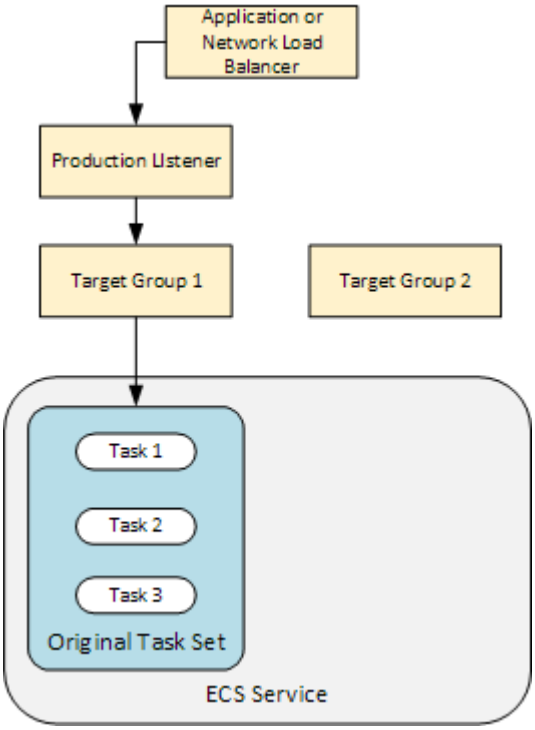
Um dein GitHub Repository zu löschen

Weitere Informationen finden Sie in der GitHub Hilfe unter Löschen eines Repositorys.

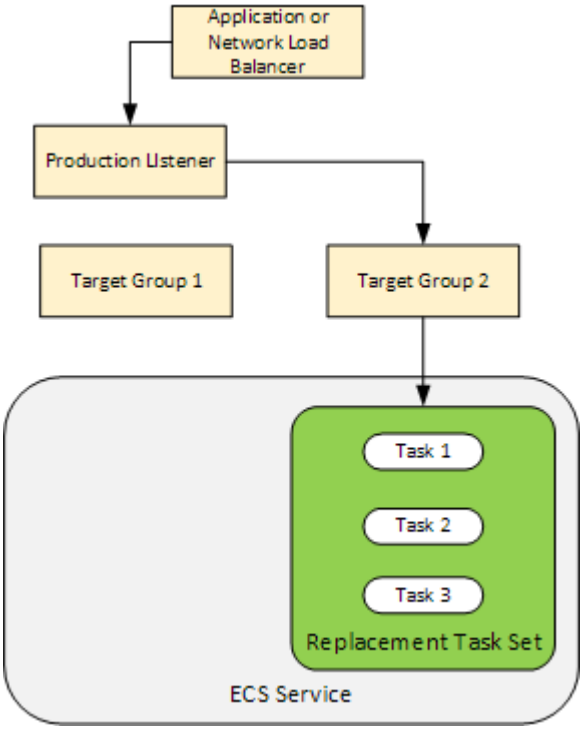
Tutorial: Bereitstellen einer Anwendung in Amazon ECS

In diesem Tutorial erfahren Sie, wie Sie eine Anwendung in Amazon ECS mithilfe von bereitstellen CodeDeploy. Sie beginnen mit einer Anwendung, die Sie bereits erstellt und in Amazon ECS bereitgestellt haben. Der erste Schritt besteht darin, Ihre Anwendung zu aktualisieren, indem Sie ihre Aufgabendefinitionsdatei mit einem neuen Tag ändern. Als Nächstes verwenden Sie, CodeDeploy um das Update bereitzustellen. CodeDeploy Installiert Ihr Update während der Bereitstellung in einem neuen Ersatz-Taskset. Anschließend verlagert es den Produktionsdatenverkehr von der ursprünglichen Version Ihrer Amazon ECS-Anwendung, die sich in ihrem ursprünglichen Task-Set befindet, auf die aktualisierte Version im Ersatz-Task-Set.

CodeDeploy Verwendet während einer Amazon ECS-Bereitstellung einen Load Balancer, der für zwei Zielgruppen und einen Production-Traffic-Listener konfiguriert ist. Das folgende Diagramm zeigt, wie der Load Balancer, der Produktions-Listener, die Zielgruppen und Ihre Amazon ECS-Anwendung zusammenhängen, bevor die Bereitstellung beginnt. In diesem Tutorial wird ein Application Load Balancer verwendet. Sie können auch einen Network Load Balancer verwenden.



Nach einer erfolgreichen Bereitstellung leitet der Produktionsdatenverkehr-Listener Datenverkehr an Ihren neuen Ersatzaufgabensatz weiter und der ursprüngliche Aufgabensatz wird beendet. Das folgende Diagramm zeigt, wie Ihre Ressourcen nach einer erfolgreichen Bereitstellung miteinander verbunden sind. Weitere Informationen finden Sie unter Was passiert während einer Amazon ECS-Bereitstellung.



Informationen zur Verwendung von AWS CLI zum Bereitstellen einer Anwendung in Amazon ECS finden Sie unter Tutorial: Einen Service mithilfe einer blue/green Bereitstellung erstellen. Informationen zur Erkennung und automatischen Bereitstellung von Änderungen an einem Amazon ECS-Service finden Sie unter Tutorial: Eine Pipeline mit einer Amazon ECR-Quelle und ECS-to-CodeDeploy Bereitstellung erstellen. CodePipeline CodeDeploy

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie die von Ihnen erstellte CodeDeploy Anwendung und Bereitstellungsgruppe verwenden, um einen Validierungstest für die Bereitstellung hinzuzufügen. Tutorial: Bereitstellen eines Amazon ECS-Service mit einem Validierungstest

Themen

- Voraussetzungen
- Schritt 1: Aktualisieren Sie Ihre Amazon ECS-Anwendung
- Schritt 2: Erstellen Sie die Datei AppSpec
- Schritt 3: Verwenden Sie die CodeDeploy Konsole, um Ihre Anwendung bereitzustellen
- Schritt 4: Bereinigen

Voraussetzungen

Um dieses Tutorial abzuschließen, müssen Sie zunächst:

- Die Schritte 2 und 3 in Erste Schritte mit CodeDeploy ausführen.
- Erstellen Sie einen Application Load Balancer, der mit zwei Zielgruppen und einem Listener konfiguriert ist. Weitere Informationen zum Erstellen eines Load Balancers mithilfe der Konsole finden Sie unter Einen Load Balancer, Zielgruppen und Listener für CodeDeploy Amazon ECS-Bereitstellungen einrichten. Informationen zum Erstellen eines Load Balancers mit dem AWS CLI finden Sie unter Schritt 1: Einen Application Load Balancer erstellen im Amazon Elastic Container Service-Benutzerhandbuch. Wenn Sie Ihren Load Balancer erstellen, notieren Sie sich Folgendes für dieses Tutorial:
  - Den Namen Ihres Load Balancer.
  - Die Namen Ihrer Zielgruppen.
  - Den Port, der vom Listener Ihres Load Balancers verwendet wird.
- Erstellen Sie einen Amazon ECS-Cluster und -Service. Weitere Informationen finden Sie in den Schritten 2, 3 und 4 im Tutorial: Einen Service mithilfe einer blue/green Bereitstellung erstellen im Amazon Elastic Container Service-Benutzerhandbuch. Notieren Sie sich Folgendes für dieses Tutorial:
  - Der Name Ihres Amazon ECS-Clusters.
  - Der ARN der Aufgabendefinition, die von Ihrem Amazon ECS-Service verwendet wird.
  - Der Name des Containers, der von Ihrem Amazon ECS-Service verwendet wird.
- Erstellen Sie einen Amazon S3 S3-Bucket für Ihre AppSpec Datei.

Schritt 1: Aktualisieren Sie Ihre Amazon ECS-Anwendung

In diesem Abschnitt aktualisieren Sie Ihre Amazon ECS-Anwendung mit einer neuen Version ihrer Aufgabendefinition. Die aktualisierte Revision fügt ein neues Schlüssel- und Tag-Paar hinzu. In Schritt 3: Verwenden Sie die CodeDeploy Konsole, um Ihre Anwendung bereitzustellen stellen Sie die aktualisierte Version Ihrer Amazon ECS-Anwendung bereit.

So aktualisieren Sie Ihre Aufgabendefinition

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/ecs/v2.
2. Wählen Sie im Navigationsbereich Task Definitions aus.
3. Wählen Sie die Aufgabendefinition aus, die von Ihrem Amazon ECS-Service verwendet wird.
4. Markieren Sie die Aufgabendefinitionsversion und wählen Sie dann Neue Version erstellen, Neue Version erstellen.

5. Nehmen Sie für dieses Tutorial eine kleine Aktualisierung der Aufgabendefinition vor, indem Sie ein Tag hinzufügen. Erstellen Sie unten auf der Seite unter Tags ein neues Tag, indem Sie ein neues Schlüssel-Wert-Paar eingeben.
6. Wählen Sie Erstellen aus.

   Die Revisionsnummer Ihrer Aufgabendefinition wird um eins erhöht.
7. Wählen Sie den Tab JSON. Notieren Sie sich Folgendes, da Sie diese Informationen im nächsten Schritt benötigen.

   - Den Wert für `taskDefinitionArn`. Das Format ist `arn:aws:ecs:`*`aws-region`*`:`*`account-id`*`:task-definition/`*`task-definition-family`*`:`*`task-definition-revision`*. Dies ist der ARN Ihrer aktualisierten Aufgabendefinition.
   - Den Wert für `name` im `containerDefinitions`-Element. Dies ist der Name Ihres Containers.
   - Den Wert für `containerPort` im `portMappings`-Element. Dies ist der Port für Ihren Container.

Schritt 2: Erstellen Sie die Datei AppSpec

In diesem Abschnitt erstellen Sie Ihre AppSpec Datei und laden sie in den Amazon S3 S3-Bucket hoch, den Sie in dem Voraussetzungen Abschnitt erstellt haben. Die AppSpec Datei für eine Amazon ECS-Bereitstellung spezifiziert Ihre Aufgabendefinition, Ihren Container-Namen und Ihren Container-Port. Weitere Informationen erhalten Sie unter AppSpec Dateibeispiel für eine Amazon ECS-Bereitstellung und AppSpec Abschnitt „Ressourcen“ für Amazon ECS-Bereitstellungen.

Um Ihre AppSpec Datei zu erstellen

1. Wenn Sie Ihre AppSpec Datei mit YAML erstellen möchten, erstellen Sie eine Datei mit dem Namen`appspec.yml`. Wenn Sie Ihre AppSpec Datei mit JSON erstellen möchten, erstellen Sie eine Datei mit dem Namen`appspec.json`.
2. Wählen Sie die entsprechende Registerkarte, je nachdem, ob Sie YAML oder JSON für Ihre AppSpec Datei verwenden, und kopieren Sie deren Inhalt in die AppSpec Datei, die Sie gerade erstellt haben. Verwenden Sie für die `TaskDefinition`-Eigenschaft den Aufgabendefinitions-ARN, den Sie im Schritt 1: Aktualisieren Sie Ihre Amazon ECS-Anwendung-Abschnitt notiert haben.

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**
>
> Ihr neuer Aufgabensatz erbt das Subnetz, die Sicherheitsgruppe, die Plattformversion und die zugewiesenen öffentlichen IP-Werte von Ihrem ursprünglichen Aufgabensatz. Sie können diese Werte für Ihren Ersatz-Tasksatz überschreiben, indem Sie ihre optionalen Eigenschaften in Ihrer AppSpec Datei festlegen. Weitere Informationen erhalten Sie

unter AppSpec Abschnitt „Ressourcen“ für Amazon ECS-Bereitstellungen und AppSpec Dateibeispiel für eine Amazon ECS-Bereitstellung.

3. Laden Sie Ihre AppSpec Datei in den S3-Bucket hoch, den Sie als Voraussetzung für dieses Tutorial erstellt haben.

Schritt 3: Verwenden Sie die CodeDeploy Konsole, um Ihre Anwendung bereitzustellen

In diesem Abschnitt erstellen Sie eine CodeDeploy Anwendungs- und Bereitstellungsgruppe, um Ihre aktualisierte Anwendung in Amazon ECS bereitzustellen. CodeDeploy Verschiebt während der Bereitstellung den Produktionsdatenverkehr für Ihre Anwendung auf die neue Version in einem neuen Ersatzaufgabensatz. Um diesen Schritt abzuschließen, benötigen Sie die folgenden Elemente:

- Ihr Amazon ECS-Clustername.
- Ihr Amazon ECS-Servicename.
- Ihr Application Load Balancer Balancer-Name.
- Ihren Produktions-Listener-Port.
- Ihre Zielgruppennamen.
- Den Namen des S3-Buckets, den Sie erstellt haben.

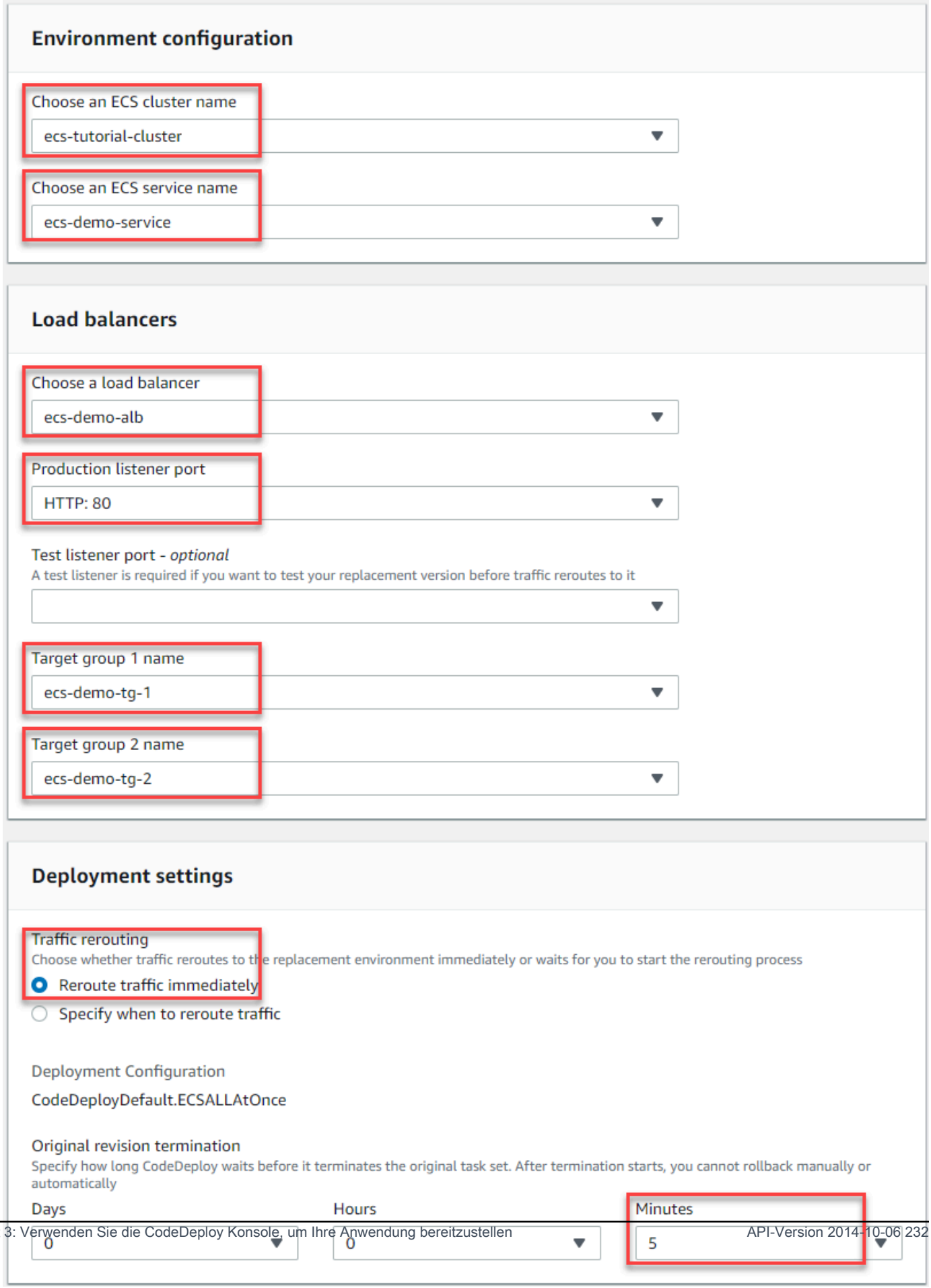
Um eine Anwendung zu erstellen CodeDeploy

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy/.
2. Wählen Sie Create application aus.
3. Geben Sie unter Application name (Anwendungsname) **`ecs-demo-codedeploy-app`** ein.
4. Wählen Sie unter Compute Platform (Datenverarbeitungsplattform) die Option Amazon ECS aus.
5. Wählen Sie Create application aus.

Um eine CodeDeploy Bereitstellungsgruppe zu erstellen

1. Wählen Sie auf der Registerkarte Deployment groups (Bereitstellungsgruppen) Ihrer Anwendungsseite Create deployment group (Bereitstellungsgruppe erstellen) aus.

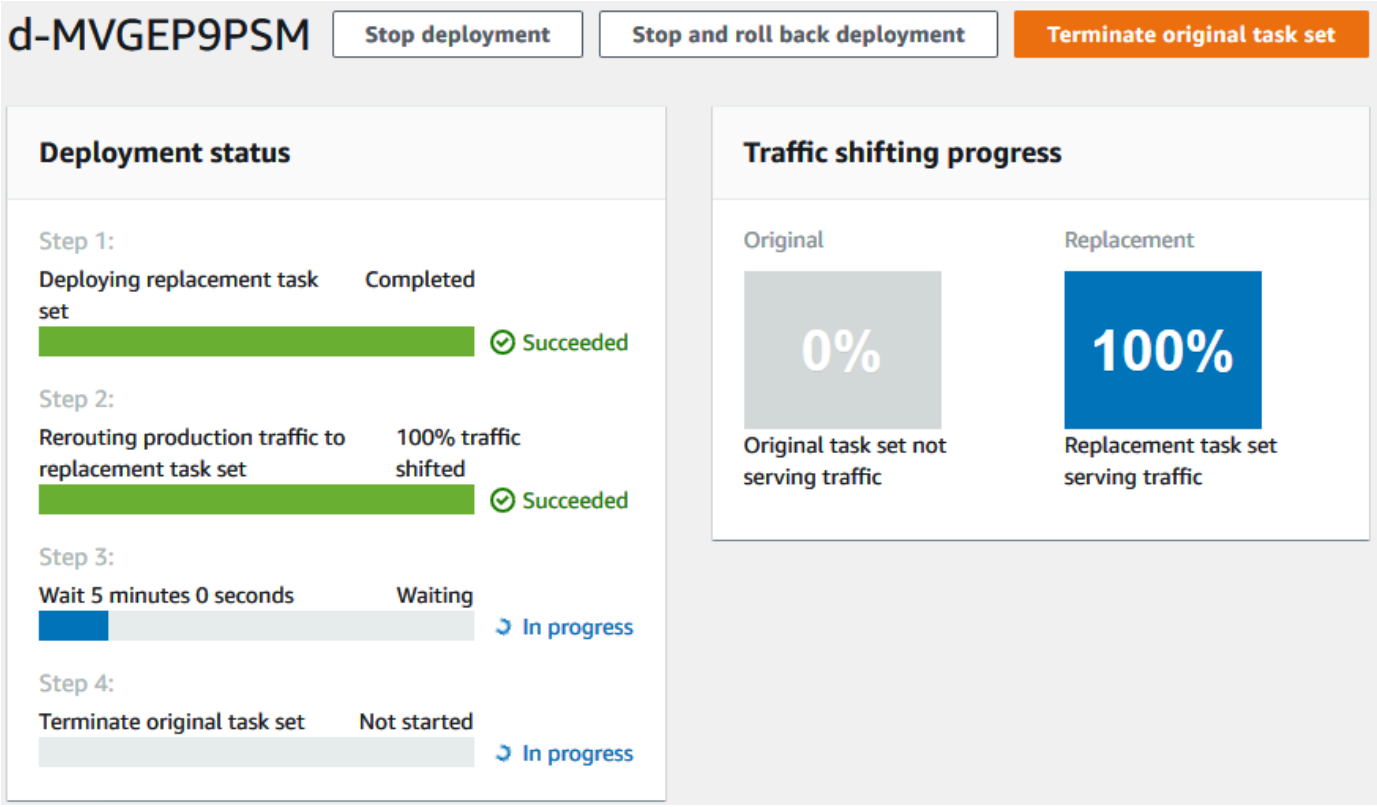
2. Geben Sie unter Deployment group name (Name der Bereitstellungsgruppe) **`ecs-demo-dg`** ein.
3. Wählen Sie unter Servicerolle eine Servicerolle aus, die CodeDeploy Zugriff auf Amazon ECS gewährt. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für AWS CodeDeploy.
4. Wählen Sie in der Umgebungskonfiguration Ihren Amazon ECS-Clusternamen und Servicenamen aus.
5. Wählen Sie unter Load Balancers den Namen des Load Balancers aus, der den Datenverkehr an Ihren Amazon ECS-Service weiterleitet.
6. Wählen Sie unter Production Listener Port den Port und das Protokoll für den Listener aus, der Produktionsdatenverkehr für Ihren Amazon ECS-Service bereitstellt (z. B. HTTP: 80). Dieses Tutorial enthält keinen optionalen Test-Listener. Wählen Sie daher keinen Port aus Test listener port (Test-Listener-Port) aus.
7. Wählen Sie unter Target group 1 name (Name der Zielgruppe 1) und Target group 2 name (Name der Zielgruppe 2) zwei verschiedene Zielgruppen aus, um den Datenverkehr während der Bereitstellung weiterzuleiten. Stellen Sie sicher, dass es sich hierbei um die Zielgruppen handelt, die Sie für Ihren Load Balancer erstellt haben. Es spielt keine Rolle, welche für Zielgruppe 1 und welche für Zielgruppe 2 verwendet wird.
8. Wählen Sie Reroute traffic immediately (Datenverkehr sofort umleiten) aus.
9. Wählen Sie für Original revision termination (Beendigung der ursprünglichen Revision) 0 Tage, 0 Stunden und 5 Minuten aus. Auf diese Weise können Sie Ihre Bereitstellung schneller abschließen, als wenn Sie die Standardeinstellung (1 Stunde) verwenden.


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. Wählen Si Create deployment group (Bereitstellungsgruppe erstellen).

Um Ihre Amazon ECS-Anwendung bereitzustellen

1. Wählen Sie auf der Konsolenseite Ihrer Bereitstellungsgruppe Create deployment (Bereitstellung erstellen) aus.
2. Wählen Sie für Deployment group (Bereitstellungsgruppe) die Option ecs-demo-dg aus.
3. Wählen Sie unter Revision type (Revisionstyp) die Option My application is stored in Amazon S3 (Meine Anwendung ist in Amazon S3 gespeichert) aus. Geben Sie unter Revision location (Speicherort der Revision) den Namen Ihres S3-Buckets ein.
4. Wählen Sie für Revision file type (Revisionsdateityp) die Option .json oder .yaml nach Bedarf aus.
5. (Optional) Geben Sie im Feld Deployment description (Bereitstellungsbeschreibung) eine Beschreibung für Ihre Bereitstellung ein.
6. Wählen Sie Create deployment.
7. Unter Deployment status (Bereitstellungsstatus) können Sie Ihre Bereitstellung überwachen. Nachdem der gesamte Produktionsdatenverkehr an das Ersatz-Task-Set weitergeleitet wurde und die Wartezeit von fünf Minuten abläuft, können Sie „Ursprünglichen Aufgabensatz beenden“ wählen, um den ursprünglichen Tasksatz sofort zu beenden. Wenn Sie Terminate original task set (Ursprünglichen Aufgabensatz beenden) nicht auswählen, wird der ursprüngliche Aufgabensatz nach der von Ihnen angegebenen fünfminütigen Wartezeit beendet.



Schritt 4: Bereinigen

Das nächste Tutorial, Tutorial: Bereitstellen eines Amazon ECS-Service mit einem Validierungstest, baut auf diesem Tutorial auf und verwendet die von Ihnen erstellte CodeDeploy -Anwendung und -Bereitstellungsgruppe. Wenn Sie die Schritte in diesem Tutorial ausführen möchten, überspringen Sie diesen Schritt und löschen Sie nicht die Ressourcen, die Sie erstellt haben.

> **Note**
>
> Für Ihr AWS Konto fallen keine Gebühren für die von Ihnen erstellten CodeDeploy Ressourcen an.

Bei den Ressourcennamen in diesen Schritten handelt es sich um die in diesem Tutorial vorgeschlagenen Namen (z. B. **`ecs-demo-codedeploy-app`** für den Namen Ihrer CodeDeploy Anwendung). Wenn Sie andere Namen verwendet haben, stellen Sie sicher, dass Sie diese während der Bereinigung verwenden.

1. Verwenden Sie den Befehl delete-deployment-group, um die Bereitstellungsgruppe zu löschen. CodeDeploy

```
aws deploy delete-deployment-group --application-name ecs-demo-codedeploy-app --
deployment-group-name ecs-demo-dg --region aws-region-id
```

2. Verwenden Sie den Befehl delete-application, um die Anwendung zu löschen. CodeDeploy

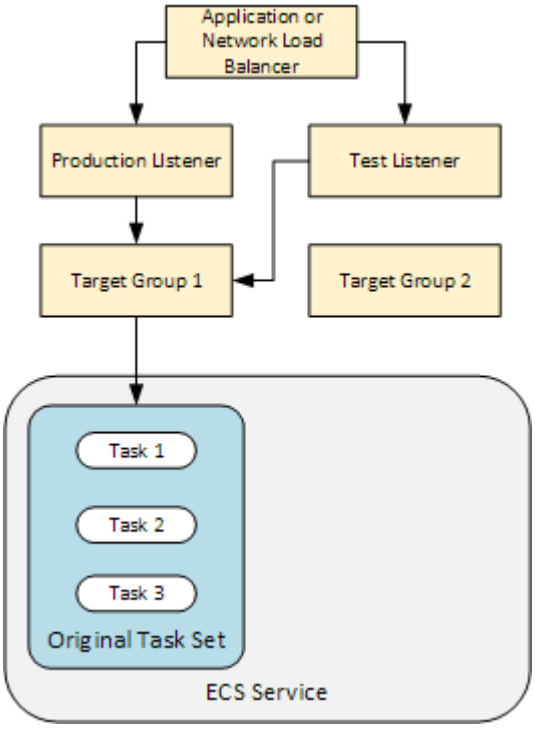
```
aws deploy delete-application --application-name ecs-demo-codedeploy-app --
region aws-region-id
```

Tutorial: Bereitstellen eines Amazon ECS-Service mit einem Validierungstest

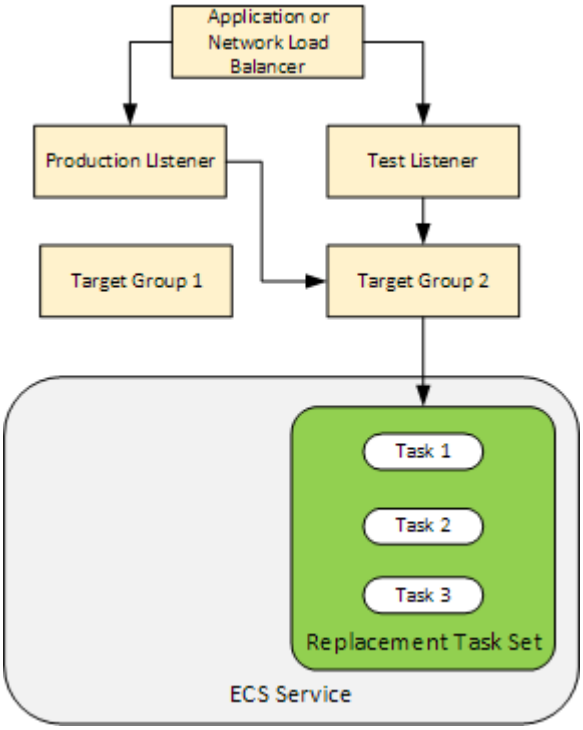
In diesem Tutorial erfahren Sie, wie Sie eine Lambda-Funktion verwenden, um einen Teil der Bereitstellung einer aktualisierten Amazon ECS-Anwendung zu validieren. In diesem Tutorial werden die CodeDeploy Anwendung, die CodeDeploy Bereitstellungsgruppe und die Amazon ECS-Anwendung verwendet, die Sie in verwendet habenTutorial: Bereitstellen einer Anwendung in Amazon ECS. Schließen Sie dieses Tutorial ab, bevor Sie mit diesem beginnen.

Um einen Validierungstest hinzuzufügen, implementieren Sie den Test zunächst in einer Lambda-Funktion. Als Nächstes geben Sie in Ihrer AppSpec Bereitstellungsdatei die Lambda-Funktion für den Lifecycle-Hook an, den Sie testen möchten. Wenn ein Validierungstest fehlschlägt, wird die Bereitstellung gestoppt, zurückgesetzt und als fehlgeschlagen gekennzeichnet. Wenn der Test erfolgreich ist, wird die Bereitstellung mit dem nächsten Bereitstellungslebenszyklus-Ereignis oder -Hook fortgesetzt.

CodeDeploy Verwendet während einer Amazon ECS-Bereitstellung mit Validierungstests einen Load Balancer, der für zwei Zielgruppen konfiguriert ist: einen Produktions-Traffic-Listener und einen Test-Traffic-Listener. Das folgende Diagramm zeigt, wie der Load Balancer, die Produktions- und Test-Listener, die Zielgruppen und Ihre Amazon ECS-Anwendung zusammenhängen, bevor die Bereitstellung beginnt. In diesem Tutorial wird ein Application Load Balancer verwendet. Sie können auch einen Network Load Balancer verwenden.



Während einer Amazon ECS-Bereitstellung gibt es fünf Lifecycle-Hooks zum Testen. Dieses Tutorial implementiert einen Test während des dritten Lebenszyklusbereitstellungs-Hooks, `AfterAllowTestTraffic`. Weitere Informationen finden Sie unter Liste der Lifecycle-Event-Hooks für eine Amazon ECS-Bereitstellung. Nach einer erfolgreichen Bereitstellung leitet der Produktionsdatenverkehr-Listener Datenverkehr an Ihren neuen Aufgabensatz weiter und der ursprüngliche Aufgabensatz wird beendet. Das folgende Diagramm zeigt, wie Ihre Ressourcen nach einer erfolgreichen Bereitstellung miteinander verbunden sind. Weitere Informationen finden Sie unter Was passiert während einer Amazon ECS-Bereitstellung.



> **Note**
>
> Wenn Sie dieses Tutorial abschließen, kann dies zu Belastungen Ihres Kontos führen. AWS Dazu gehören mögliche Gebühren für CodeDeploy AWS Lambda, und CloudWatch. Weitere Informationen finden Sie unter AWS CodeDeploy Preise, AWS Lambda Preise und CloudWatch Amazon-Preise.

Themen

- Voraussetzungen
- Schritt 1: Erstellen Sie einen Test-Listener
- Schritt 2: Aktualisieren Sie Ihre Amazon ECS-Anwendung
- Schritt 3: Erstellen Sie eine Lifecycle-Hook-Lambda-Funktion
- Schritt 4: Aktualisieren Sie Ihre AppSpec Datei
- Schritt 5: Verwenden Sie die CodeDeploy Konsole, um Ihren Amazon ECS-Service bereitzustellen
- Schritt 6: Zeigen Sie die Ausgabe Ihrer Lambda-Hook-Funktion in Logs an CloudWatch
- Schritt 7: Bereinigen

Voraussetzungen

Um dieses Tutorial erfolgreich abzuschließen, müssen Sie zunächst:

- Die Voraussetzungen unter Voraussetzungen für Tutorial: Bereitstellen einer Anwendung in Amazon ECS erfüllen.
- Führen Sie die Schritte unter Tutorial: Bereitstellen einer Anwendung in Amazon ECS aus. Notieren Sie sich Folgendes:
  - Den Namen Ihres Load Balancer.
  - Die Namen Ihrer Zielgruppen.
  - Den Port, der vom Listener Ihres Load Balancers verwendet wird.
  - Den ARN Ihres Load Balancers. Damit erstellen Sie einen neuen Listener.
  - Den ARN einer Ihrer Zielgruppen. Damit erstellen Sie einen neuen Listener.
  - Die CodeDeploy Anwendung und die Bereitstellungsgruppe, die Sie erstellen.
  - Die von Ihnen erstellte AppSpec Datei, die von Ihrer CodeDeploy Bereitstellung verwendet wird. Sie bearbeiten diese Datei in diesem Tutorial.

Schritt 1: Erstellen Sie einen Test-Listener

Eine Amazon ECS-Bereitstellung mit Validierungstests erfordert einen zweiten Listener. Dieser Listener wird verwendet, um Testdatenverkehr für Ihre aktualisierte Amazon ECS-Anwendung in einem Ersatzaufgabensatz bereitzustellen. Ihre Validierungstests werden anhand des Testdatenverkehrs ausgeführt.

Der Listener für Ihren Testdatenverkehr kann eine der Zielgruppen verwenden. Verwenden Sie den AWS CLI Befehl create-listener, um einen zweiten Listener mit einer Standardregel zu erstellen, die Testdatenverkehr an Port 8080 weiterleitet. Verwenden Sie den ARN Ihres Load Balancers und den ARN einer Ihrer Zielgruppen.

```
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
```

Schritt 2: Aktualisieren Sie Ihre Amazon ECS-Anwendung

In diesem Abschnitt aktualisieren Sie Ihre Amazon ECS-Anwendung, sodass sie eine neue Version ihrer Aufgabendefinition verwendet. Sie erstellen die neue Revision und fügen ihr eine kleinere Aktualisierung hinzu, indem Sie ein Tag hinzufügen.

So aktualisieren Sie Ihre Aufgabendefinition

1. Öffnen Sie die klassische Amazon ECS-Konsole unter https://console.aws.amazon.com/ecs/.
2. Wählen Sie im Navigationsbereich Task Definitions aus.
3. Aktivieren Sie das Kontrollkästchen für die Aufgabendefinition, die von Ihrem Amazon ECS-Service verwendet wird.
4. Wählen Sie Create new revision (Neue Revision erstellen) aus.
5. Nehmen Sie eine kleine Aktualisierung der Aufgabendefinition vor, indem Sie ein Tag hinzufügen. Erstellen Sie unten auf der Seite unter Tags ein neues Tag, indem Sie ein neues Schlüssel- und Wert-Paar eingeben.
6. Wählen Sie Erstellen aus. Sie sollten sehen, dass die Versionsnummer Ihrer Aufgabendefinition um eins erhöht wurde.
7. Wählen Sie den Tab JSON. Notieren Sie sich den Wert für `taskDefinitionArn`. Das Format ist `arn:aws:ecs:`*`aws-region`*`:` *`account-id`*`:task-definition/`*`task-definition-family`*`:` *`task-definition-revision`*. Dies ist der ARN Ihrer aktualisierten Aufgabendefinition.

Schritt 3: Erstellen Sie eine Lifecycle-Hook-Lambda-Funktion

In diesem Abschnitt implementieren Sie eine Lambda-Funktion für den `AfterAllowTestTraffic` Hook Ihrer Amazon ECS-Bereitstellung. Die Lambda-Funktion führt einen Validierungstest durch, bevor die aktualisierte Amazon ECS-Anwendung installiert wird. Für dieses Tutorial kehrt die Lambda-Funktion zurück`Succeeded`. Während einer realen Bereitstellung geben Validierungstests `Succeeded` oder `Failed` zurück, abhängig vom Ergebnis des Validierungstests. Während einer realen Bereitstellung können Sie auch eine Lambda-Testfunktion für einen oder mehrere der anderen Event-Hooks (`BeforeInstall`, `AfterInstallBeforeAllowTraffic`, und`AfterAllowTraffic`) des Amazon ECS-Bereitstellungslebenszyklus implementieren. Weitere Informationen finden Sie unter Liste der Lifecycle-Event-Hooks für eine Amazon ECS-Bereitstellung.

Eine IAM-Rolle ist erforderlich, um Ihre Lambda-Funktion zu erstellen. Die Rolle gewährt der Lambda-Funktion die Berechtigung, in CloudWatch Logs zu schreiben und den Status eines CodeDeploy Lifecycle-Hooks festzulegen.

So erstellen Sie eine IAM-Rolle

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen) aus.
3. Erstellen Sie eine Rolle mit den folgenden Eigenschaften:

   - Trusted entity (Vertrauenswürdige Entität): AWS Lambda.
   - Berechtigungen: AWSLambdaBasicExecutionRole. Dadurch wird Ihrer Lambda-Funktion die Berechtigung erteilt, in CloudWatch Logs zu schreiben.
   - Role name (Name der Rolle): **`lambda-cli-hook-role`**.

   Weitere Informationen finden Sie unter Eine AWS Lambda Ausführungsrolle erstellen.
4. Fügen Sie die Berechtigung `codedeploy:PutLifecycleEventHookExecutionStatus` an die Rolle an, die Sie erstellt haben. Dadurch erhalten Ihre Lambda-Funktionen die Berechtigung, den Status eines CodeDeploy Lifecycle-Hooks während einer Bereitstellung festzulegen. Weitere Informationen finden Sie unter Hinzufügen von IAM-Identitätsberechtigungen im AWS Identity and Access Management Benutzerhandbuch und PutLifecycleEventHookExecutionStatusin der CodeDeploy API-Referenz.

Um eine **`AfterAllowTestTraffic`**Hook-Lambda-Funktion zu erstellen

1. Erstellen Sie eine Datei mit dem Namen `AfterAllowTestTraffic.js` und dem folgenden Inhalt.

```
'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. Erstellen Sie ein Lambda-Bereitstellungspaket.

```
zip AfterAllowTestTraffic.zip AfterAllowTestTraffic.js
```

3. Verwenden Sie den `create-function` Befehl, um eine Lambda-Funktion für Ihren `AfterAllowTestTraffic` Hook zu erstellen.

```
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. Notieren Sie sich den ARN Ihrer Lambda-Funktion in der `create-function` Antwort. Sie verwenden diesen ARN, wenn Sie die AppSpec Datei Ihrer CodeDeploy Bereitstellung im nächsten Schritt aktualisieren.

Schritt 4: Aktualisieren Sie Ihre AppSpec Datei

In diesem Abschnitt aktualisieren Sie Ihre AppSpec Datei mit einem `Hooks` Abschnitt. In dem `Hooks` Abschnitt geben Sie eine Lambda-Funktion für den `AfterAllowTestTraffic` Lifecycle-Hook an.

Um Ihre AppSpec Datei zu aktualisieren

1. Öffnen Sie die AppSpec Dateidatei, die Sie in Schritt 2: Erstellen Sie die Datei AppSpec der erstellt habenTutorial: Bereitstellen einer Anwendung in Amazon ECS.
2. Aktualisieren Sie die `TaskDefinition`-Eigenschaft mit dem Aufgabendefinitions-ARN, den Sie sich in Schritt 2: Aktualisieren Sie Ihre Amazon ECS-Anwendung notiert haben.
3. Kopieren Sie den `Hooks` Abschnitt und fügen Sie ihn in Ihre AppSpec Dateidatei ein. Aktualisieren Sie den ARN anschließend `AfterAllowTestTraffic` mit dem ARN der Lambda-Funktion, die Sie notiert Schritt 3: Erstellen Sie eine Lifecycle-Hook-Lambda-Funktion haben.

   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. Speichern Sie Ihre AppSpec Datei und laden Sie sie in den zugehörigen S3-Bucket hoch.

Schritt 5: Verwenden Sie die CodeDeploy Konsole, um Ihren Amazon ECS-Service bereitzustellen

In diesem Abschnitt aktualisieren Sie Ihre Bereitstellungsgruppe, indem Sie den Port für Ihren Test-Listener angeben. Dies ist der Listener, den Sie in Schritt 1: Erstellen Sie einen Test-Listener erstellt haben. Während der Bereitstellung führt CodeDeploy Ihren Validierungstest während des `AfterAllowTestTraffic`-Bereitstellungslebenszyklus-Hooks mithilfe von Testdatenverkehr aus, der an Ihren neuen Aufgabensatz mit dem Test-Listener geleitet wird. Ihr Validierungstest gibt das Ergebnis `Succeeded` zurück, sodass die Bereitstellung mit dem nächsten Bereitstellungslebenszyklusereignis fortgesetzt wird. In einem realen Szenario gibt Ihre Testfunktion `Succeeded` oder `Failed` zurück.

So fügen Sie Ihrer Bereitstellungsgruppe einen Test-Listener hinzu

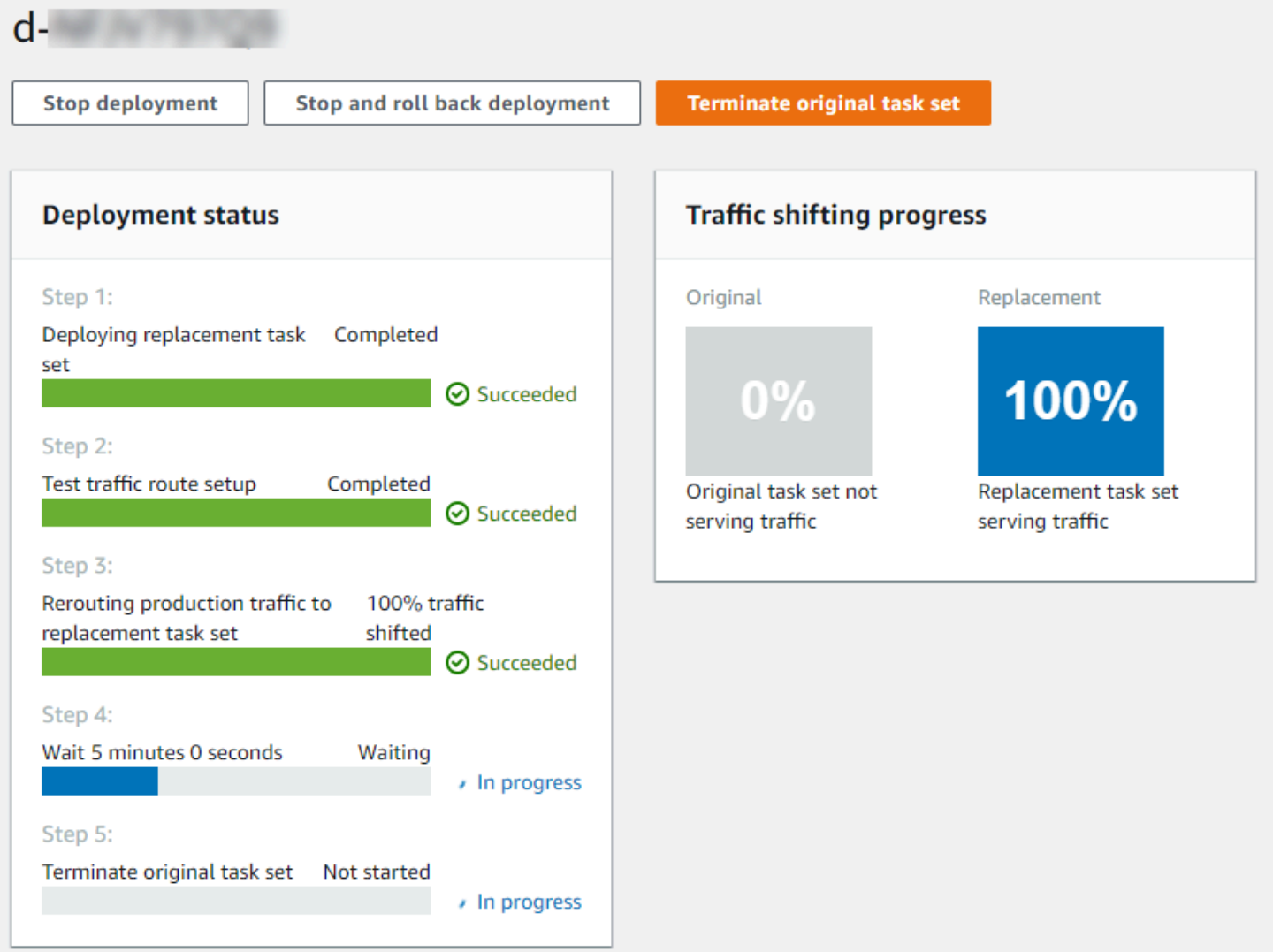
1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy/.
2. Wählen Sie im Navigationsbereich Applications (Anwendungen) aus.
3. Wählen Sie die Anwendung aus, die Sie in Tutorial: Bereitstellen einer Anwendung in Amazon ECS erstellt haben. Wenn Sie den vorgeschlagenen Namen verwendet haben, lautet er ecs-demo-codedeploy-app.
4. Wählen Sie unter Deployment groups (Bereitstellungsgruppen) die Bereitstellungsgruppe aus, die Sie in Tutorial: Bereitstellen einer Anwendung in Amazon ECS erstellt haben. Wenn Sie den vorgeschlagenen Namen verwendet haben, lautet er ecs-demo-dg.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie unter Test listener port (Test-Listener-Port) den Port und das Protokoll für den Test-Listener aus, den Sie zuvor in diesem Tutorial erstellt haben. Dies sollte HTTP: 8080 sein.
7. Wählen Sie Änderungen speichern aus.

Um Ihre Amazon ECS-Anwendung bereitzustellen

1. Wählen Sie auf der Konsolenseite Ihrer Bereitstellungsgruppe Create deployment (Bereitstellung erstellen) aus.
2. Wählen Sie für Deployment group (Bereitstellungsgruppe) die Option ecs-demo-dg aus.
3. Wählen Sie unter Revision type (Revisionstyp) die Option My application is stored in Amazon S3 (Meine Anwendung ist in Amazon S3 gespeichert) aus. Geben Sie unter Revisionsort den Namen Ihres S3-Buckets und der AppSpec Datei ein (z. B.**`s3://my-s3-bucket/appspec.json`**).
4. Wählen Sie für Revision file type (Revisionsdateityp) die Option .json oder .yaml aus.
5. (Optional) Geben Sie im Feld Deployment description (Bereitstellungsbeschreibung) eine Beschreibung für Ihre Bereitstellung ein.
6. Wählen Sie Create deployment.

Sie können Ihre Bereitstellung im Deployment status (Bereitstellungsstatus) überwachen. Nachdem der gesamte Produktionsdatenverkehr an den Ersatz-Tasksatz weitergeleitet wurde, können Sie „Ursprünglichen Tasksatz beenden“ wählen, um den ursprünglichen Tasksatz sofort zu beenden. Wenn Sie Terminate original task set (Urprünglichen Aufgabensatz beenden) nicht auswählen,

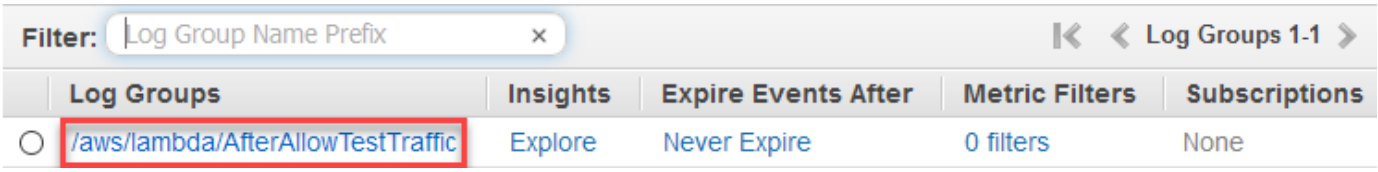
wird der ursprüngliche Aufgabensatz nach der Dauer beendet, die Sie beim Erstellen Ihrer Bereitstellungsgruppe angegeben haben.



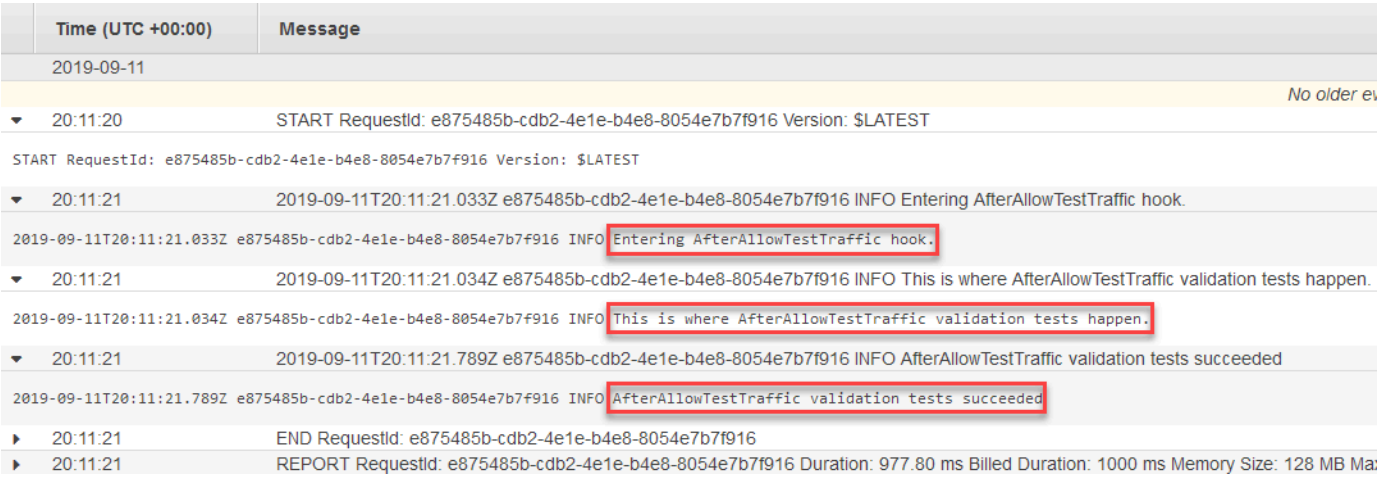
Schritt 6: Zeigen Sie die Ausgabe Ihrer Lambda-Hook-Funktion in Logs an CloudWatch

Wenn Ihre CodeDeploy Bereitstellung erfolgreich ist, sind auch die Validierungstests in Ihren Lambda-Hook-Funktionen erfolgreich. Sie können dies überprüfen, indem Sie im Protokoll unter Logs nach der Hook-Funktion suchen. CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Logs (Protokolle) aus. Sie sollten eine neue Protokollgruppe für die Lambda-Hook-Funktion sehen, die Sie in Ihrer AppSpec Datei angegeben haben.



3. Wählen Sie die neue Protokollgruppe aus. Das sollte/aws/lambda/seinAfterAllowTestTrafficHook.
4. Wählen Sie den Protokollstream. Wenn Sie mehr als einen Protokoll-Stream sehen, wählen Sie unter Last Event Time (Letzte Ereigniszeit) denjenigen mit dem neuesten Datum und der aktuellen Uhrzeit aus.
5. Erweitern Sie die Protokollstream-Ereignisse, um zu bestätigen, dass Ihre Lambda-Hook-Funktion Erfolgsmeldungen in das Protokoll geschrieben hat. Das Folgende zeigt, dass die `AfterAllowTraffic` Lambda-Hook-Funktion erfolgreich war.



Schritt 7: Bereinigen

Wenn Sie dieses Tutorial abgeschlossen haben, sollten Sie die damit verknüpften Ressourcen bereinigen, um zu vermeiden, dass für nicht verwendete Ressourcen Kosten entstehen. Bei den Ressourcennamen in diesem Schritt handelt es sich um die in diesem Tutorial vorgeschlagenen Namen (z. B. **`ecs-demo-codedeploy-app`** für den Namen Ihrer CodeDeploy Anwendung). Wenn Sie verschiedene Namen verwendet haben, stellen Sie sicher, dass Sie diese in Ihrer Bereinigung verwenden.

So bereinigen Sie die Tutorial-Ressourcen

1. Verwenden Sie den Befehl delete-deployment-group, um die Bereitstellungsgruppe zu löschen. CodeDeploy

```
aws deploy delete-deployment-group --application-name ecs-demo-deployment-group --
deployment-group-name ecs-demo-dg --region aws-region-id
```

2. Verwenden Sie den Befehl delete-application, um die Anwendung zu löschen. CodeDeploy

```
aws deploy delete-application --application-name ecs-demo-deployment-group --
region aws-region-id
```

3. Verwenden Sie den Befehl delete-function, um Ihre Lambda-Hook-Funktion zu löschen.

```
aws lambda delete-function --function-name AfterAllowTestTraffic
```

4. Verwenden Sie den Befehl delete-log-group, um Ihre Protokollgruppe zu löschen. CloudWatch

```
aws logs delete-log-group --log-group-name /aws/lambda/AfterAllowTestTraffic
```

Tutorial: Stellen Sie eine aktualisierte Lambda-Funktion bereit mit CodeDeploy und AWS Serverloses Anwendungsmodell

AWS SAM ist ein Open-Source-Framework für die Erstellung serverloser Anwendungen. Es transformiert und erweitert die YAML-Syntax in einer AWS SAM Vorlage in eine CloudFormation Syntax, um serverlose Anwendungen wie eine Lambda-Funktion zu erstellen. Weitere Informationen finden Sie unter Was ist das serverlose Anwendungsmodell? AWS

In diesem Tutorial verwenden Sie AWS SAM, um eine Lösung zu erstellen, die Folgendes bietet:

- Erzeugt Ihre Lambda-Funktion.
- Erstellt Ihre CodeDeploy Anwendung und Bereitstellungsgruppe.
- Erstellt zwei Lambda-Funktionen, die Bereitstellungsvalidierungstests während CodeDeploy Lifecycle-Hooks ausführen.
- Erkennt, wenn Ihre Lambda-Funktion aktualisiert wird. Die Aktualisierung der Lambda-Funktion löst eine Bereitstellung aus CodeDeploy , bei der der Produktionsverkehr schrittweise von der ursprünglichen Version Ihrer Lambda-Funktion auf die aktualisierte Version verlagert wird.

> **Note**
>
> In diesem Tutorial müssen Sie Ressourcen erstellen, die möglicherweise Kosten für Ihr AWS -Konto verursachen. Dazu gehören mögliche Gebühren für CodeDeploy Amazon CloudWatch und AWS Lambda. Weitere Informationen finden Sie unter CodeDeploy Preise, CloudWatch Amazon-Preise und AWS Lambda Preise.

Themen

- Voraussetzungen
- Schritt 1: Richten Sie Ihre Infrastruktur ein
- Schritt 2: Aktualisieren Sie die Lambda-Funktion
- Schritt 3: Stellen Sie die aktualisierte Lambda-Funktion bereit
- Schritt 4: Zeigen Sie Ihre Bereitstellungsergebnisse an
- Schritt 5: Bereinigen

Voraussetzungen

Um dieses Tutorial abzuschließen, müssen Sie zunächst:

- Führen Sie die Schritte unter Erste Schritte mit CodeDeploy aus.
- Installieren Sie die AWS Serverless Application Model CLI. Weitere Informationen finden Sie unter AWS SAM-CLI installieren.
- Erstellen Sie einen S3-Bucket. AWS SAM lädt die Artefakte, auf die in Ihrer AWS SAM-Vorlage verwiesen wird, in diesen Bucket hoch.

Schritt 1: Richten Sie Ihre Infrastruktur ein

In diesem Thema erfahren Sie, wie Sie Dateien für Ihre AWS SAM Vorlage und Ihre Lambda-Funktionen erstellen. AWS SAM Anschließend verwenden Sie die `deploy` Befehle AWS SAM `package` und, um die Komponenten in Ihrer Infrastruktur zu generieren. Wenn Ihre Infrastruktur bereit ist, haben Sie eine CodeDeploy Anwendungs- und Bereitstellungsgruppe, die Lambda-Funktion zum Aktualisieren und Bereitstellen sowie zwei Lambda-Funktionen, die Validierungstests enthalten, die ausgeführt werden, wenn Sie die Lambda-Funktion bereitstellen. Wenn Sie fertig sind,

können CloudFormation Sie Ihre Komponenten in der Lambda-Konsole anzeigen oder Ihre Lambda-Funktion testen. AWS CLI

Themen

- Erstellen Sie Ihre Dateien
- Package Sie das AWS SAM-Anwendung
- Stellen Sie das bereit AWS SAM-Anwendung
- (Optional) inspizieren und testen Sie Ihre Infrastruktur

Erstellen Sie Ihre Dateien

Um Ihre Infrastruktur zu erstellen, müssen Sie die folgenden Dateien erstellen:

- `template.yml`
- `myDateTimeFunction.js`
- `beforeAllowTraffic.js`
- `afterAllowTraffic.js`

Themen

- Erstelle deine AWS SAM-Vorlage
- Erstellen Sie eine Datei für Ihre Lambda-Funktion
- Erstellen Sie eine Datei für Ihre BeforeAllowTraffic Lambda-Funktion
- Erstellen Sie eine Datei für Ihre AfterAllowTraffic Lambda-Funktion

Erstelle deine AWS SAM-Vorlage

Erstellen Sie eine AWS SAM-Vorlagendatei, die die Komponenten in Ihrer Infrastruktur spezifiziert.

Um Ihre zu erstellen AWS SAM-Vorlage

1. Erstellen Sie ein Verzeichnis mit dem Namen `SAM-Tutorial`.
2. Erstellen Sie in Ihrem Verzeichnis `SAM-Tutorial` eine Datei namens `template.yml`.
3. Kopieren Sie den folgenden YAML-Code in `template.yml`. Dies ist Ihre AWS SAM -Vorlage.

```
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
```

Diese Vorlage gibt Folgendes an. Weitere Informationen finden Sie unter AWS SAM Vorlagenkonzepte.

Eine Lambda-Funktion namens **`myDateTimeFunction`**

Wenn diese Lambda-Funktion veröffentlicht wird, verknüpft sie die `AutoPublishAlias` Zeile in der Vorlage mit einem Alias namens`live`. Später in diesem Tutorial löst ein Update dieser Funktion eine Bereitstellung aus, bei der der Produktionsdatenverkehr schrittweise von der Originalversion auf die aktualisierte Version verlagert wird. AWS CodeDeploy

Zwei Validierungsfunktionen für die Lambda-Bereitstellung

Die folgenden Lambda-Funktionen werden während CodeDeploy Lifecycle-Hooks ausgeführt. Die Funktionen enthalten Code, der die Bereitstellung der aktualisierten `myDateTimeFunction` validiert. Das Ergebnis der Validierungstests wird mithilfe der `PutLifecycleEventHookExecutionStatus`-API-Methode an CodeDeploy übergeben. Wenn ein Validierungstest fehlschlägt, schlägt die Bereitstellung fehl und wird zurückgesetzt.

- `CodeDeployHook_beforeAllowTraffic` wird während des `BeforeAllowTraffic`-Hooks ausgeführt.
- `CodeDeployHook_afterAllowTraffic` wird während des `AfterAllowTraffic`-Hooks ausgeführt.

Der Name beider Funktionen beginnt mit `CodeDeployHook_`. Die `CodeDeployRoleForLambda` Rolle erlaubt Aufrufe der `invoke` Lambda-Methode nur in Lambda-Funktionen, deren Namen mit diesem Präfix beginnen. Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“ für einen AWS Lambda-Bereitstellung und PutLifecycleEventHookExecutionStatus in der CodeDeploy -API-Referenz.

Automatische Erkennung einer aktualisierten Lambda-Funktion

Der `AutoPublishAlias`-Begriff weist das Framework an, zu erkennen, wenn sich die `myDateTimeFunction`-Funktion ändert, und sie dann mit dem `live`-Alias bereitzustellen.

Eine Bereitstellungskonfiguration

Die Bereitstellungskonfiguration bestimmt die Geschwindigkeit, mit der Ihre CodeDeploy Anwendung den Datenverkehr von der ursprünglichen Version der Lambda-Funktion auf die neue Version verlagert. Diese Vorlage gibt die vordefinierte Bereitstellungskonfiguration `Linear10PercentEvery1Minute` an.

> **Note**
>
> Sie können keine benutzerdefinierte Bereitstellungskonfiguration in einer AWS SAM-Vorlage angeben. Weitere Informationen finden Sie unter Create a Deployment Configuration.

Funktionen von Bereitstellungslebenszyklus-Hooks

Der `Hooks`-Abschnitt gibt die Funktionen an, die während Lebenszyklusereignis-Hooks ausgeführt werden. `PreTraffic` gibt die Funktion an, die während des `BeforeAllowTraffic`-Hooks ausgeführt wird. `PostTraffic` gibt die Funktion an, die während des `AfterAllowTraffic`-Hooks ausgeführt wird.

Berechtigungen für Lambda zum Aufrufen einer anderen Lambda-Funktion

Die angegebene `lambda:InvokeFunction` Berechtigung gewährt der von der AWS SAM-Anwendung verwendeten Rolle die Berechtigung, eine Lambda-Funktion aufzurufen. Dies ist erforderlich, wenn die `CodeDeployHook_afterAllowTraffic` Funktionen `CodeDeployHook_beforeAllowTraffic` und die bereitgestellte Lambda-Funktion während der Validierungstests aufrufen.

Erstellen Sie eine Datei für Ihre Lambda-Funktion

Erstellen Sie die Datei für die Funktion, die Sie später in diesem Tutorial aktualisieren und bereitstellen.

> **Note**
>
> Eine Lambda-Funktion kann jede Laufzeit verwenden, die von AWS Lambda unterstützt wird. Weitere Informationen finden Sie unter AWS Lambda -Laufzeiten.

Um Ihre Lambda-Funktion zu erstellen

1. Erstellen Sie eine Textdatei und speichern Sie sie als `myDateTimeFunction.js` im `SAM-Tutorial`-Verzeichnis.
2. Kopieren Sie den folgenden Node.js Code in`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;
      }
    };
```

Die Lambda-Funktion gibt den Tag, den Monat und das Jahr für gestern, heute oder morgen zurück. Später in diesem Tutorial entfernen Sie die Kommentarzeichen von Code, der die Funktion aktualisiert, um Informationen über den Tag oder die Uhrzeit zurückzugeben, die Sie angeben (z. B. Tag, Monat und Jahr oder die aktuelle Stunde, Minute und Sekunde). Das von erstellte Framework AWS SAM erkennt die aktualisierte Version der Funktion und stellt sie bereit.

> **Note**
>
> Diese Lambda-Funktion wird auch in einem AWS Cloud9 Tutorial verwendet. AWS Cloud9 ist eine cloudbasierte integrierte Entwicklungsumgebung. Informationen zum Erstellen, Ausführen, Aktualisieren und Debuggen dieser Funktion finden Sie unter AWS Lambda Tutorial für AWS Cloud9. AWS Cloud9

Erstellen Sie eine Datei für Ihre BeforeAllowTraffic Lambda-Funktion

Erstellen Sie die Datei für Ihre `beforeAllowTraffic` Hook-Lambda-Funktion.

1. Erstellen Sie eine Textdatei und speichern Sie sie als `beforeAllowTraffic.js` im `SAM-Tutorial`-Verzeichnis.
2. Kopieren Sie den folgenden Node.js Code in`beforeAllowTraffic.js`. Diese Funktion wird während des `BeforeAllowTraffic`-Hooks Ihrer Bereitstellung ausgeführt.

```
'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");
        }
       });
      }
     });
    }
```

Erstellen Sie eine Datei für Ihre AfterAllowTraffic Lambda-Funktion

Erstellen Sie die Datei für Ihre `afterAllowTraffic` Hook-Lambda-Funktion.

1. Erstellen Sie eine Textdatei und speichern Sie sie als `afterAllowTraffic.js` im `SAM-Tutorial`-Verzeichnis.
2. Kopieren Sie den folgenden Node.js Code in`afterAllowTraffic.js`. Diese Funktion wird während des `AfterAllowTraffic`-Hooks Ihrer Bereitstellung ausgeführt.

```
'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 Sie das AWS SAM-Anwendung

Sie sollten nun vier Dateien in Ihrem `SAM-Tutorial`-Verzeichnis haben:

- `beforeAllowTraffic.js`
- `afterAllowTraffic.js`
- `myDateTimeFunction.js`
- `template.yml`

Sie können jetzt den AWS sam package SAM-Befehl verwenden, um Artefakte für Ihre Lambda-Funktionen und CodeDeploy -Anwendungen zu erstellen und zu verpacken. Die Artefakte werden in einen S3-Bucket hochgeladen. Die Ausgabe des Befehls ist eine neue Datei mit dem Namen `package.yml`. Diese Datei wird im nächsten Schritt vom AWS sam deploy SAM-Befehl verwendet.

> **Note**
>
> Weitere Informationen zu diesem sam package Befehl finden Sie in der AWS SAM CLI-Befehlsreferenz im AWS Serverless Application Model Entwicklerhandbuch.

Führen Sie Folgendes im `SAM-Tutorial`-Verzeichnis aus.

```
sam package \
  --template-file template.yml \
  --output-template-file package.yml \
  --s3-bucket amzn-s3-demo-bucket
```

Geben Sie für den `s3-bucket` Parameter den Amazon S3 S3-Bucket an, den Sie als Voraussetzung für dieses Tutorial erstellt haben. Der `output-template-file` gibt den Namen der neuen Datei an, die vom AWS sam deploy SAM-Befehl verwendet wird.

Stellen Sie das bereit AWS SAM-Anwendung

Verwenden Sie den AWS sam deploy SAM-Befehl mit der `package.yml` Datei, um Ihre Lambda-Funktionen sowie Ihre CodeDeploy Anwendungs- und Bereitstellungsgruppe mithilfe von CloudFormation zu erstellen.

> **Note**
>
> Weitere Informationen zu diesem sam deploy Befehl finden Sie in der AWS SAM CLI-Befehlsreferenz im AWS Serverless Application Model Entwicklerhandbuch.

Führen Sie folgenden Befehl im `SAM-Tutorial`-Verzeichnis aus.

```
sam deploy \
  --template-file package.yml \
  --stack-name my-date-time-app \
  --capabilities CAPABILITY_IAM
```

Der `--capabilities CAPABILITY_IAM` Parameter ist erforderlich, um die Erstellung von CloudFormation IAM-Rollen zu autorisieren.

(Optional) inspizieren und testen Sie Ihre Infrastruktur

In diesem Thema wird gezeigt, wie Sie Ihre Infrastrukturkomponenten anzeigen und Ihre Lambda-Funktion testen können.

Um das Ergebnis Ihres Stacks zu sehen, nachdem Sie **`Sam`** Deploy ausgeführt haben

1. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Klicken Sie im Navigationsbereich auf Stacks. Der `my-date-time-app`-Stack wird oben angezeigt.
3. Wählen Sie die Registerkarte Events (Ereignisse) aus, um zu sehen, welche Ereignisse abgeschlossen sind. Sie können die Ereignisse während der Stack-Erstellung anzeigen. Wenn die Erstellung des Stacks abgeschlossen ist, werden alle Stack-Erstellungsereignisse angezeigt.
4. Klicken Sie bei ausgewähltem Stack auf Resources (Ressourcen). In der Spalte Typ können Sie Ihre Lambda-Funktionen, `myDateTimeFunctionCodeDeployHook_beforeAllowTraffic`, und `CodeDeployHook_afterAllowTraffic` sehen. Die Spalte Physikalische ID jeder Ihrer Lambda-Funktionen enthält einen Link zum Anzeigen der Funktionen in der Lambda-Konsole.

> **Note**
>
> Dem Namen der `myDateTimeFunction` Lambda-Funktion wird der Name des CloudFormation Stacks vorangestellt und es wird ein Bezeichner hinzugefügt, also sieht es so aus. `my-date-time-app-myDateTimeFunction-123456ABCDEF`

5. Öffnen Sie die Konsole unter CodeDeploy . https://console.aws.amazon.com/codedeploy/
6. Erweitern Sie im Navigationsbereich Deploy (Bereitstellen) und klicken Sie danach auf Applications (Anwendungen).
7. Sie sollten eine neue CodeDeploy Anwendung sehen, die von CloudFormation erstellt wurde und deren Name mit beginnt`my-date-time-app-ServerlessDeploymentApplication`. Wählen Sie diese Anwendung aus.
8. Sie sollten eine Bereitstellungsgruppe mit einem Namen sehen, der mit `my-date-time-app-myDateTimeFunctionDeploymentGroup` beginnt. Wählen Sie diese Bereitstellungsgruppe aus.

   Unter Bereitstellungskonfiguration sollte Folgendes angezeigt werden CodeDeployDefault.LambdaLinear10PercentEvery1Minute.

(Optional), um Ihre Funktion zu testen (Konsole)

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie im Navigationsbereich Ihre `my-date-time-app-myDateTimeFunction`-Funktion aus. In der Konsole enthält der Name eine Kennung, sodass er wie `my-date-time-app-myDateTimeFunction-123456ABCDEF` aussieht.
3. Wählen Sie Test aus.
4. Geben Sie unter Event name (Ereignisname) einen Namen für Ihr Testereignis ein.
5. Geben Sie Folgendes für Ihr Testereignis ein und wählen Sie dann Create (Erstellen) aus.

```
{
  "option": "date",
  "period": "today"
}
```

6. Wählen Sie Test aus. Sie sollten nur Ihr Testereignis in der Liste der Testereignisse sehen.

Für Execution result (Ausführungsergebnis) sollten Sie erfolgreich sehen.

7. Erweitern Sie unter Execution result (Ausführungsergebnis) die Option Details, um die Ergebnisse anzuzeigen. Sie sollten den aktuellen Monat, Tag und das Jahr sehen.

(Optional), um Ihre Funktion zu testen (AWS CLI)

1. Suchen Sie den ARN Ihrer Lambda-Funktion. Es wird oben in der Lambda-Konsole angezeigt, wenn Sie Ihre Funktion aufrufen.
2. Führen Sie den folgenden Befehl aus. *your-function-arn*Ersetzen Sie durch die Funktion ARN.

```
aws lambda invoke \
--function your-function-arn \
--cli-binary-format raw-in-base64-out \
--payload "{\"option\": \"date\", \"period\": \"today\"}" out.txt
```

3. Öffnen Sie `out.txt`, um sicherzustellen, dass das Ergebnis den aktuellen Monat, Tag und das Jahr enthält.

Schritt 2: Aktualisieren Sie die Lambda-Funktion

In diesem Thema aktualisieren Sie Ihre `myDateTimeFunction.js`-Datei. Im nächsten Schritt verwenden Sie die Datei, um die aktualisierte Funktion bereitzustellen. Dies löst CodeDeploy die Bereitstellung aus, indem der Produktionsdatenverkehr von der aktuellen Version der Lambda-Funktion auf die aktualisierte Version verlagert wird.

Um Ihre Lambda-Funktion zu aktualisieren

1. Öffnen Sie `myDateTimeFunction.js`.
2. Entfernen Sie die beiden Kommentarmarkierungen ("`/*`" und "`*/`") und den Erläuterungstext am Anfang und am Ende des `case` mit dem Namen `time` im `switch`-Block.

   Mit dem nicht kommentierten Code können Sie einen neuen Parameter, `time`, an die Funktion übergeben. Wenn Sie `time` an die aktualisierte Funktion übergeben, wird die aktuelle `hour`, `minute` und `second` zurückgegeben.
3. Speichern `myDateTimeFunction.js`. Sie sollte wie folgt aussehen:

```
'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;
  }
};
```

Schritt 3: Stellen Sie die aktualisierte Lambda-Funktion bereit

In diesem Schritt verwenden Sie Ihr Update, `myDateTimeFunction.js` um Ihre Lambda-Funktion zu aktualisieren und deren Bereitstellung zu initiieren. Sie können den Bereitstellungsfortschritt in der CodeDeploy AWS Lambda OR-Konsole überwachen.

Die `AutoPublishAlias: live` Zeile in Ihrer AWS SAM Vorlage veranlasst Ihre Infrastruktur, Aktualisierungen von Funktionen zu erkennen, die den `live` Alias verwenden. Ein Update Ihrer Funktion löst eine Bereitstellung aus CodeDeploy , indem der Produktionsdatenverkehr von der ursprünglichen Version der Funktion auf die aktualisierte Version verlagert wird.

Die sam deploy Befehle sam package und werden verwendet, um die Bereitstellung Ihrer Lambda-Funktion zu aktualisieren und auszulösen. Sie haben diese Befehle in Package Sie das AWS SAM-Anwendung und Stellen Sie das bereit AWS SAM-Anwendung ausgeführt.

So stellen Sie Ihre aktualisierte Lambda-Funktion bereit

1. Führen Sie folgenden Befehl im `SAM-Tutorial`-Verzeichnis aus.

   ```
   sam package \
     --template-file template.yml \
     --output-template-file package.yml  \
     --s3-bucket amzn-s3-demo-bucket
   ```

   Dadurch wird ein neuer Satz von Artefakten erstellt, die auf Ihre aktualisierte Lambda-Funktion in Ihrem S3-Bucket verweisen.

2. Führen Sie folgenden Befehl im `SAM-Tutorial`-Verzeichnis aus.

   ```
   sam deploy \
     --template-file package.yml \
     --stack-name my-date-time-app \
   ```

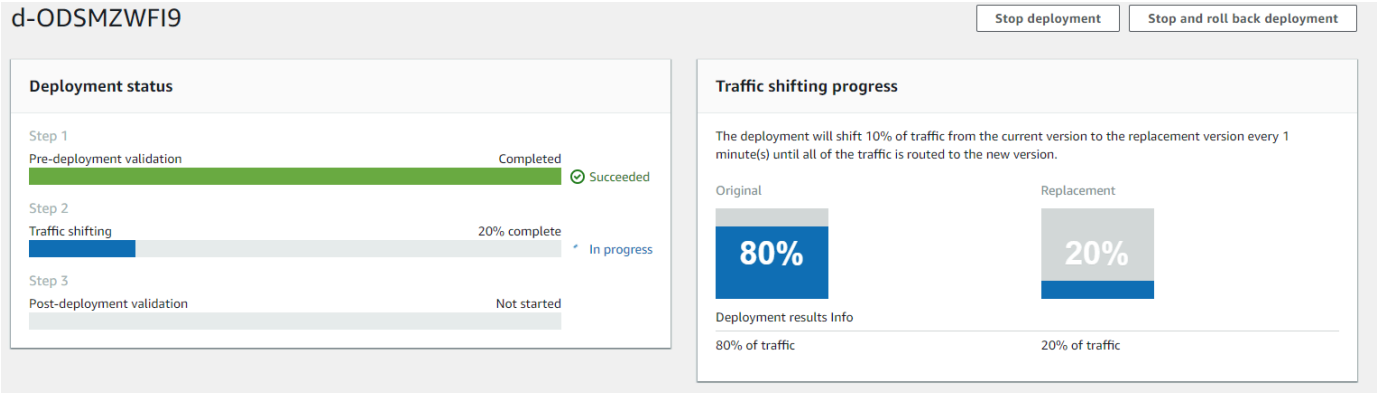
```
  --capabilities CAPABILITY_IAM
```

Da der Stack-Name immer noch lautet, wird CloudFormation erkanntmy-date-time-app, dass es sich um ein Stack-Update handelt. Um Ihren aktualisierten Stack anzuzeigen, kehren Sie zur CloudFormation Konsole zurück und wählen Sie im Navigationsbereich Stacks aus.

(Optional), um den Datenverkehr während einer Bereitstellung anzuzeigen (CodeDeploy Konsole)

1. Öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy/.
2. Erweitern Sie im Navigationsbereich Anwendungen und wählen Sie dann Ihre my-date-time-app-Anwendung aus. ServerlessDeploymentApplication
3. Wählen Sie unter Deployment groups (Bereitstellungsgruppen) die Bereitstellungsgruppe Ihrer Anwendung aus. Der Status sollte In progress (In Bearbeitung) sein.
4. Wählen Sie unter Deployment group history (Bereitstellungsgruppenverlauf) die Bereitstellung aus, die gerade ausgeführt wird.

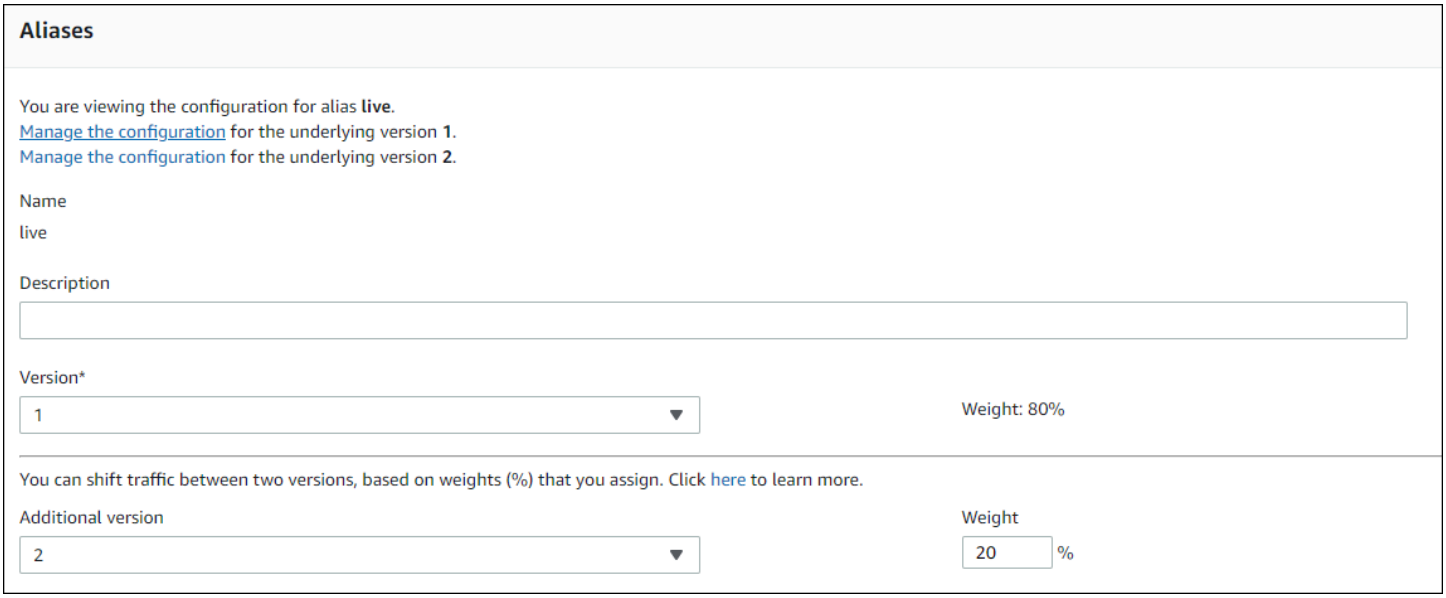
   Die Fortschrittsanzeige Traffic Shifting (Verschiebung von Datenverkehr) und die Prozentsätze in den Feldern Original und Ersatz auf dieser Seite zeigen ihren Fortschritt an.



(Optional) zur Anzeige des Datenverkehrs während einer Bereitstellung (Lambda-Konsole)

1. Öffnen Sie die AWS Lambda Konsole unter. https://console.aws.amazon.com/lambda/
2. Wählen Sie im Navigationsbereich Ihre my-date-time-app-myDateTimeFunction-Funktion aus. In der Konsole enthält der Name eine Kennung, sodass er wie my-date-time-app-myDateTimeFunction-123456ABCDEF aussieht.
3. Wählen Sie Aliase und dann Live aus.

Die Gewichtungen neben Ihrer ursprünglichen Funktionsversion (Version 1) und Ihrer aktualisierten Funktionsversion (Version 2) zeigen an, wie viel Datenverkehr an jede Version zum Zeitpunkt des Ladens dieser AWS Lambda -Konsolenseite geleitet wird. Die Seite aktualisiert die Gewichtungen nicht im Laufe der Zeit. Wenn Sie die Seite einmal pro Minute aktualisieren, verringert sich die Gewichtung für Version 1 um 10 Prozent und die Gewichtung für Version 2 erhöht sich um 10 Prozent, bis die Gewichtung für Version 2 100 beträgt.



Schritt 4: Zeigen Sie Ihre Bereitstellungsergebnisse an

In diesem Schritt zeigen Sie die Ergebnisse Ihrer Bereitstellung an. Wenn Ihre Bereitstellung erfolgreich ist, können Sie bestätigen, dass Ihre aktualisierte Lambda-Funktion Produktionsdatenverkehr empfängt. Wenn Ihre Bereitstellung fehlschlägt, können Sie mithilfe von CloudWatch Logs die Ergebnisse der Validierungstests in der Lambda-Funktion anzeigen, die während der Lifecycle-Hooks Ihrer Bereitstellung ausgeführt werden.

Themen

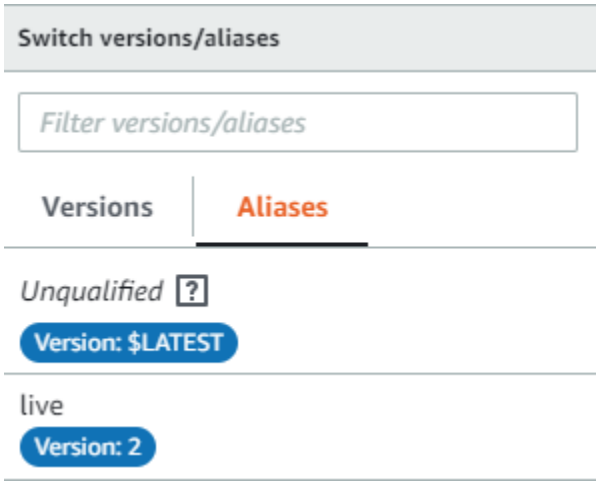
- Testen Sie Ihre bereitgestellte Funktion
- Hook-Ereignisse in CloudWatch Logs anzeigen

Testen Sie Ihre bereitgestellte Funktion

Der sam deploy Befehl aktualisiert die `my-date-time-app-myDateTimeFunction` Lambda-Funktion. Die Funktionsversion wird auf 2 aktualisiert und dem `live`-Alias hinzugefügt.

Um das Update in der Lambda-Konsole zu sehen

1. Öffnen Sie die AWS Lambda Konsole unter. https://console.aws.amazon.com/lambda/
2. Wählen Sie im Navigationsbereich die `my-date-time-app-myDateTimeFunction`-Funktion aus. In der Konsole enthält der Name eine Kennung, sodass er wie `my-date-time-app-myDateTimeFunction-123456ABCDEF` aussieht.
3. Wählen Sie Qualifiers (Qualifizierer) und dann Aliases (Aliasse) aus. Nachdem die Bereitstellung abgeschlossen ist (ca. 10 Minuten), sollte für den `live`-Alias Version: 2 angezeigt werden.



4. Unter Function code (Funktionscode) können Sie den Quellcode für Ihre Funktion anzeigen. Ihre Änderungen sollten angezeigt werden.
5. (Optional) Sie können die Testanweisungen in Schritt 2: Aktualisieren Sie die Lambda-Funktion verwenden, um Ihre aktualisierte Funktion zu testen. Erstellen Sie ein neues Testereignis mit der folgenden Nutzlast und stellen Sie sicher, dass das Ergebnis die aktuelle Stunde, Minute und Sekunde enthält.

```
{
    "option": "time"
  }
```

Um die aktualisierte Funktion AWS CLI zu testen, führen Sie den folgenden Befehl aus und öffnen Sie ihn, `out.txt` um zu bestätigen, dass das Ergebnis die aktuelle Stunde, Minute und Sekunde enthält.

```
aws lambda invoke --function your-function-arn --payload "{\"option\": \"time\"}"
 out.txt
```

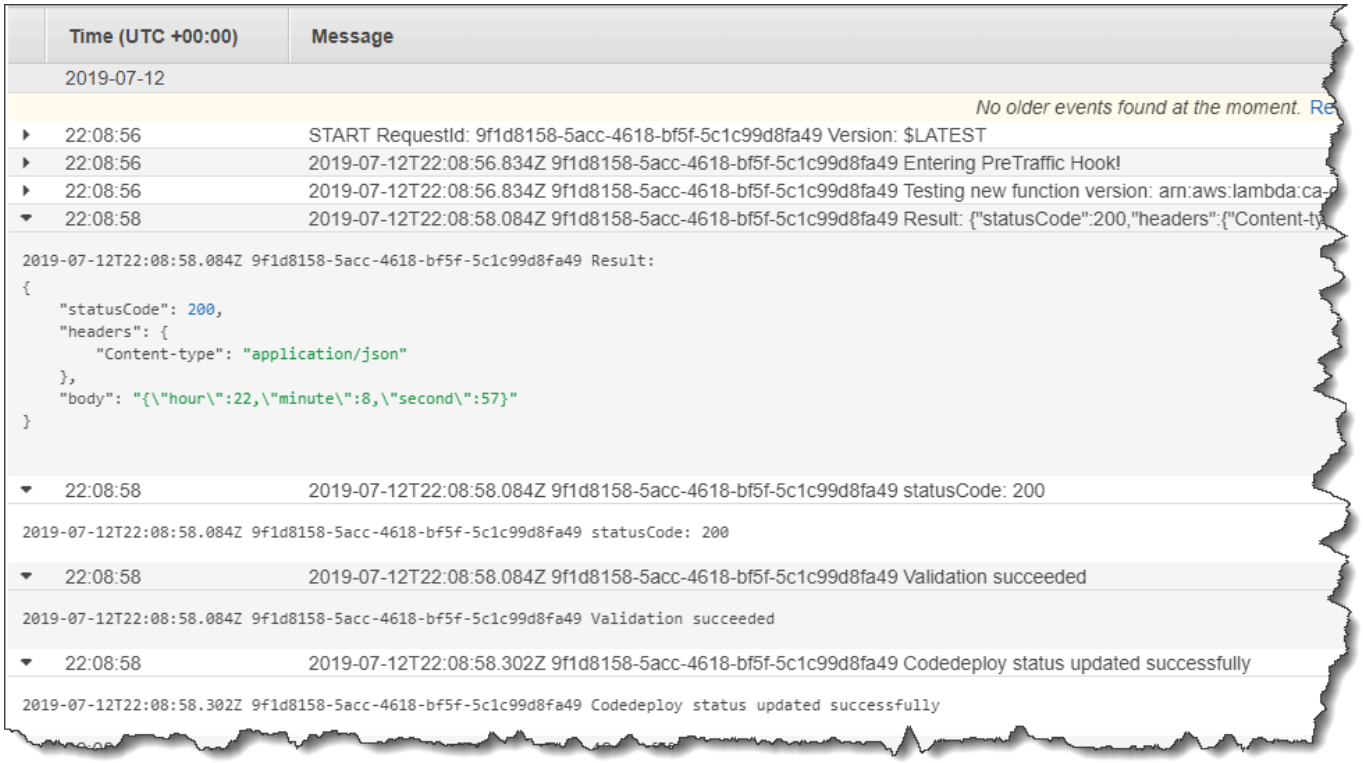
> **Note**
>
> Wenn Sie den verwenden AWS CLI , um Ihre Funktion zu testen, bevor die Bereitstellung abgeschlossen ist, erhalten Sie möglicherweise unerwartete Ergebnisse. Das liegt daran, dass jede Minute 10 Prozent des Datenverkehrs CodeDeploy schrittweise auf die aktualisierte Version umgestellt werden. Während der Bereitstellung zeigt der Datenverkehr weiterhin auf die ursprüngliche Version, sodass `aws lambda invoke` möglicherweise die Originalversion verwendet. Nach 10 Minuten ist die Bereitstellung abgeschlossen und der gesamte Datenverkehr zeigt auf die neue Version der Funktion.

Hook-Ereignisse in CloudWatch Logs anzeigen

CodeDeploy Führt während des `BeforeAllowTraffic` Hooks Ihre `CodeDeployHook_beforeAllowTraffic` Lambda-Funktion aus. CodeDeploy Führt während des `AfterAllowTraffic` Hooks Ihre `CodeDeployHook_afterAllowTraffic` Lambda-Funktion aus. Jede Funktion führt einen Validierungstest aus, der die aktualisierte Version Ihrer Funktion mit dem neuen `time`-Parameter aufruft. Wenn das Update Ihrer Lambda-Funktion erfolgreich ist, verursacht die `time` Option keinen Fehler und die Validierung ist erfolgreich. Wenn die Funktion nicht aktualisiert wurde, führt der nicht erkannte Parameter zu einem Fehler und die Validierung schlägt fehl. Diese Validierungstests dienen nur zu Demonstrationszwecken. Sie schreiben eigene Tests, um Ihre Bereitstellung zu validieren. Sie können die CloudWatch Logs-Konsole verwenden, um Ihre Validierungstests anzusehen.

Um Ihre CodeDeploy Hook-Ereignisse einzusehen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Logs (Protokolle) aus.
3. Wählen Sie in der Liste der Protokollgruppen/aws/lambda/CodeDeployHook_before AllowTraffic oder/aws/lambda/CodeDeployHookAllowTraffic_after aus.
4. Wählen Sie den Protokollstream. Sie sollten nur eine sehen.
5. Erweitern Sie die Ereignisse, um ihre Details anzuzeigen.



Schritt 5: Bereinigen

Um weitere Gebühren für Ressourcen zu vermeiden, die Sie in diesem Tutorial verwendet haben, löschen Sie die von Ihrer AWS SAM Vorlage erstellten Ressourcen und die von Ihren Lambda-Validierungsfunktionen erstellten CloudWatch Protokolle.

Um deine zu löschen CloudFormation Stack

1. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Spalte Stacks Ihren `my-date-time-app`-Stack aus und klicken Sie dann auf Löschen.
3. Wählen Sie bei Aufforderung Delete stack (Stack löschen) aus. Die Lambda-Funktionen, die CodeDeploy Anwendungs- und Bereitstellungsgruppe sowie die IAM-Rollen, die von erstellt wurden, AWS SAM werden gelöscht.

Um Ihre Logs in Logs zu löschen CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Logs (Protokolle) aus.
3. Wählen Sie in der Liste der Protokollgruppen die Schaltfläche neben/aws/lambda/CodeDeployHook_before AllowTraffic aus.
4. Wählen Sie unter Actions (Aktionen) die Option Delete log group (Protokollgruppe löschen) und dann Yes, Delete (Ja, löschen) aus.
5. Wählen Sie in der Liste der Protokollgruppen die Schaltfläche neben/aws/lambda/CodeDeployHookAllowTraffic_after aus.
6. Wählen Sie unter Actions (Aktionen) die Option Delete log group (Protokollgruppe löschen) und dann Yes, Delete (Ja, löschen) aus.

Mit dem CodeDeploy Agenten arbeiten

Der AWS CodeDeploy Agent ist ein Softwarepaket, das, wenn es auf einer Instanz installiert und konfiguriert ist, die Verwendung dieser Instanz in CodeDeploy Bereitstellungen ermöglicht.

AWS unterstützt aktiv die neueste Nebenversion des CodeDeploy Agenten. Die neueste veröffentlichte Version finden Sie unterVersionsverlauf des Agenten CodeDeploy.

> **Note**
>
> Der CodeDeploy Agent ist nur erforderlich, wenn Sie ihn auf einer EC2/On-Premises Rechenplattform bereitstellen. Der Agent ist für Bereitstellungen, die die Amazon ECS- oder AWS Lambda-Rechenplattform verwenden, nicht erforderlich.

Eine Konfigurationsdatei wird auf der Instance platziert, wenn der Agent installiert wird. Diese Datei wird verwendet, um festzulegen, wie der Agent arbeitet. Diese Konfigurationsdatei spezifiziert Verzeichnispfade und andere Einstellungen, die bei AWS CodeDeploy der Interaktion mit der Instance verwendet werden sollen. Sie können einige der Konfigurationsoptionen in der Datei ändern. Hinweise zur Arbeit mit der CodeDeploy Agent-Konfigurationsdatei finden Sie unterCodeDeploy Referenz zur Agentenkonfiguration.

Weitere Informationen zur Arbeit mit dem CodeDeploy Agenten, z. B. zu Schritten zur Installation, Aktualisierung und Überprüfung von Versionen, finden Sie unterVerwaltung des CodeDeploy Agentenbetriebs.

Themen

- Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy
- Kommunikationsprotokoll und Port für den CodeDeploy Agenten
- Versionsverlauf des Agenten CodeDeploy
- Den CodeDeploy Prozess verwalten
- Revision der Anwendung und Bereinigung der Protokolldateien
- Vom Agenten installierte Dateien CodeDeploy
- Verwaltung des CodeDeploy Agentenbetriebs

Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy

Unterstützte Amazon EC2 AMI-Betriebssysteme

Der CodeDeploy Agent wurde auf den folgenden Amazon EC2 AMI-Betriebssystemen getestet:

- Amazon Linux 2023 (x86_64, aarch64)
- Amazon Linux 2 (x86_64, aarch64)
- RedHat Enterprise Linux (RHEL) 8, 9, 10 (x86_64, aarch64)
- Oracle Linux 8, 9, 10 (x86_64, aarch64)
- Rocky Linux 9, 10 (x86_64, aarch64)
- CentOS Stream 9, 10 (x86_64, aarch64)
- SLES 15 (x86_64, aarch64)
- Debian 11, 12, 13 (x86_64, aarch64)
- Ubuntu Server 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 auf EC2-Mac-Instances mit Apple-Silizium (aarch64)

> **Note**
>
> Die folgenden Plattformen wurden nur mit der CodeDeploy Agentenversion 2.0.x und höher getestet: Windows Server 2025, RHEL 10, Oracle Linux 8, 9, 10, Rocky Linux 9, 10, CentOS Stream 9, 10, SLES 15, Debian 11, 12, 13 und Ubuntu Server 24.04, 25.04, 26.04.
> macOS wird nur auf der CodeDeploy Agentenversion 2.1.2 und höher unterstützt. Der Agent unterstützt macOS 15, 26, 27 mit Apple-Silizium.

Der CodeDeploy Agent ist als Open Source verfügbar, sodass Sie ihn an Ihre Bedürfnisse anpassen können. Es kann mit anderen Amazon EC2 AMI-Betriebssystemen verwendet werden. Weitere Informationen finden Sie im CodeDeploy Agenten-Repository unter GitHub.

Unterstützte lokale Betriebssysteme

Der CodeDeploy Agent wurde auf den folgenden lokalen Betriebssystemen getestet:

- Windows Server 2016, 2019, 2022, 2025 (x86_64)
- RedHat Enterprise Linux (RHEL) 8, 9, 10 (x86_64, aarch64)
- Oracle Linux 8, 9, 10 (x86_64, aarch64)
- Rocky Linux 9, 10 (x86_64, aarch64)
- CentOS Stream 9, 10 (x86_64, aarch64)
- SLES 15 (x86_64, aarch64)
- Debian 11, 12, 13 (x86_64, aarch64)
- Ubuntu Server 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 mit Apple-Silizium (aarch64)

> **Note**
>
> Die folgenden Plattformen wurden nur mit der CodeDeploy Agentenversion 2.0.x und höher getestet: Windows Server 2025, RHEL 10, Oracle Linux 8, 9, 10, Rocky Linux 9, 10, CentOS Stream 9, 10, SLES 15, Debian 11, 12, 13 und Ubuntu Server 24.04, 25.04, 26.04.
> macOS wird nur auf der CodeDeploy Agentenversion 2.1.2 und höher unterstützt. Der Agent unterstützt macOS 15, 26, 27 mit Apple-Silizium.

Der CodeDeploy Agent ist als Open Source verfügbar, sodass Sie ihn an Ihre Bedürfnisse anpassen können. Er kann mit anderen Betriebssystemen lokaler Instances verwendet werden. Weitere Informationen finden Sie im CodeDeploy Agenten-Repository unter GitHub.

Kommunikationsprotokoll und Port für den CodeDeploy Agenten

Der CodeDeploy Agent kommuniziert ausgehend über HTTPS über Port 443.

Wenn der CodeDeploy Agent auf einer EC2-Instance ausgeführt wird, verwendet er den EC2-Metadaten-Endpunkt, um instanzbezogene Informationen abzurufen. Weitere Informationen zum Einschränken und Gewähren von Service-Zugriff auf Instance-Metadaten finden Sie hier.

Versionsverlauf des Agenten CodeDeploy

Auf Ihren Instances muss eine unterstützte Version des CodeDeploy Agenten ausgeführt werden. Die derzeit unterstützte Mindestversion ist 2.1.x.

> **Note**
>
> Wir empfehlen, die neueste Version des Agenten zu verwenden. CodeDeploy Wenn Sie Probleme haben, aktualisieren Sie auf die neueste Version, bevor Sie sich an den AWS Support wenden. Informationen zum Upgrade finden Sie unterAktualisieren Sie den CodeDeploy Agenten.

> **Status der Einführung von Version 2.x**
>
> Version 2.x des CodeDeploy Agenten wird derzeit überall eingeführt AWS-Regionen und ist möglicherweise noch nicht in allen Regionen verfügbar. Version 2.x ist ein optionales Upgrade. Informationen zum Upgrade von älteren Versionen auf Version 2.x finden Sie unter. Installieren Sie den CodeDeploy Agenten Informationen darüber, welche Version Sie installieren können, finden Sie unter. Ermitteln Sie die neueste verfügbare Agentenversion in einem AWS-Region

In der folgenden Tabelle sind alle Versionen des CodeDeploy Agenten sowie die Funktionen und Verbesserungen aufgeführt, die in jeder Version enthalten sind.

Version	Datum der Veröffentlichung	Details
2.1.2	13. Oktober 2026	Hinzugefügt: Support für macOS 15, 26, 27 mit Apple-Sil izium. Detaillierte Anweisungen finden Sie unter Installieren Sie den CodeDeploy Agenten für macOS. Hinzugefügt: Parallele Bereichsdownloads von Anwendung srevisionen von Amazon S3, standardmäßig deaktiviert. Weitere Informationen finden Sie unter CodeDeploy Referenz zur Agentenkonfiguration.

Version	Datum der Veröffentlichung	Details
		Problem behoben: Die Ausgabe des Lifecycle-Event-Skripts wird `codedeploy-agent-deployments.log` zeilenweise geschrieben, wobei die Bereitstellungs-ID vorangestellt wird.
2.1.1	2. Oktober 2026	Problem behoben: Der Agent stellt jetzt eine Verbindung zum richtigen CodeDeploy Endpunkt in der AWS European Sovereign Cloud her.
2.1.0	7. September 2026	Hinzugefügt: Systemeigene Unterstützung für den `RESTART` Bereitstellungsmodus, bei dem die Anwendung auf den Ziel-Instances neu gestartet wird, wobei die Version aus der letzten erfolgreichen Bereitstellung der Bereitstellungsgruppe verwendet wird. Der Agent wird von der Kopie der Revision aus installiert, die sich bereits auf der Instanz befindet. Wenn diese Kopie fehlt oder unvollständig ist, lädt der Agent die Version herunter, sodass die Bereitstellung trotzdem abgeschlossen ist. Informationen zum Bereitstellungsmodus selbst finden Sie unter DeploymentMode in der AWS CodeDeploy API-Referenz. Geändert (Sicherheit): Der Agent lehnt jetzt eine Bereitstellung ab, deren AppSpec Dateipfad außerhalb des Revisionsverzeichnisses der Anwendung aufgelöst wird. Innerhalb der Revision verschachtelte Pfade, wie z. B. `configs/appspec.yml`, werden weiterhin unterstützt.

Version	Datum der Veröffentlichung	Details
2.0.1	24. August 2026	Problem behoben: Unter Windows werden AppSpec Dateipfade, die mit einem umgekehrten Schrägstrich (\) oder mit mehr als einem Trennzeichen beginnen (z. B. `//my-folder`), jetzt innerhalb der Revision aufgelöst, nicht mehr im Stammverzeichnis des Laufwerks. Beispiele für unterstützte `source` Werte finden Sie unter. AppSpec Abschnitt „Dateien“ (nur EC2/On-Premises Bereitstellungen)

Version	Datum der Veröffentlichung	Details
2.0.0	15. Juli 2026	Geändert: Der CodeDeploy Agent ist jetzt verfügbar Rust-based und wird als eine einzige eigenständige native Binärdatei ausgeliefert. Die Ruby-Laufzeitabhängigkeit wurde gelöscht. Ruby ist nicht mehr erforderlich, um den Agenten zu installieren oder auszuführen. Der Agent ist standardmäßig verhaltenskompatibel mit Version 1.8.x. Bestehende Konfigurationsdateien, AppSpec Handhabung, Lifecycle-Hook-Semantik, Festplatten-Layout und Bereitstellungsergebnisse bleiben unverändert, sofern nicht unten anders angegeben. Hinzugefügt: Erweiterte Betriebssystemunterstützung, einschließlich Windows Server 2025, RHEL 10 und Ubuntu Server 26.04 sowie zusätzliche Linux-Distributionen und ARM-Varianten (aarch64). Die vollständige Liste der mit dieser Version getesteten Plattformen finden Sie unter. Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy Hinzugefügt: Ein `codedeploy-agent update` Befehl für die Selbstaktualisierung von Agenten auf Abruf. Hinzugefügt: Ein lokaler Befehlsport. Dies ist eine optionale TCP-Verwaltungsschnittstelle, die nur die Loopback-Adresse (127.0.0.1) überwacht, um den Agentenstatus abzufragen und Befehle lokal einzufügen. Sie ist standardmäßig deaktiviert und aktiviert mit. `enable_command_port` Geändert: Der lokale Bereitstellungsbefehl ist jetzt`codedeploy-agent deploy-local` . Der `codedeploy-local` Befehl bleibt aus Gründen der Abwärtskompatibilität verfügbar . Alle Befehlsoptionen werden beibehalten, und lokale Pfade, Amazon S3 (`s3://`) und GitHub Bundle-Quellen werden weiterhin unterstützt.

Version	Datum der Veröffentlichung	Details
		Geändert: Unter Linux wird der Agent von einer nativen Systemd-Unit (`codedeploy-agent.service`) verwaltet; das SysV-Skript init.d ist nicht mehr installiert. Wird verwendet, um den Agenten `systemctl` zu verwalten. Geändert (Sicherheit): Die TLS-Zertifikatsverifizierung ist immer aktiviert und Core-Dumps sind standardmäßig deaktiviert. Hinzugefügt: Optionale Hardening-Einstellungen, die Sie aktivieren können, um Ihre eigenen Sicherheitsanforderungen zu erfüllen. Jede Einstellung ist standardmäßig deaktiviert, und Sie aktivieren die gewünschten Einstellungen einzeln. Sie können Bundles ablehnen, die symbolische Links, Pfaddurch querung oder unsichere Berechtigungen enthalten, die Umgebung von Lifecycle-Event-Hooks einschränken und die Berechtigungen von Agentenverzeichnissen und Protokoll dateien einschränken. Eine vollständige Liste finden Sie unter CodeDeploy Referenz zur Agentenkonfiguration. Geändert: Das Befehlszeilen-Installationsskript ist jetzt in Bash (früher Ruby) geschrieben. Es erfordert entweder oder. `curl` `wget` Geändert: Das Installationsskript für Version 2.0.x wird unter dem `latestv2/` Präfix in den regionalen `aws-codedeploy-` *`region-identifier`* Buckets veröffentlicht. Das `latest/` Präfix dient weiterhin dem Installationsskript für Version 1.8.x. Geändert: veröffentlicht AWS jetzt Version 2.0.x als `AWSCodeDeployAgentV2` Systems Manager Distributor-Paket. Das `AWSCodeDeployAgent` Paket unterstützt weiterhin Version 1.8.x.

Version	Datum der Veröffentlichung	Details
		Entfernt: Die `ssl_verify_peer` Konfigurationseinstellung und die ProcessManager Ruby-Konfigurationsschlüssel. Wichtig: Derzeit gibt es keinen automatischen Aktualisierungspfad von Version 1.8.x auf 2.0.0. Um auf 2.0.0 zu aktualisieren, installieren Sie es mit dem `AWSCodeDeployAgentV2` Systems Manager Distributor-Paket oder führen Sie das regionale `latestv2/install` Skript manuell aus. Detaillierte Anweisungen finden Sie unter Installieren Sie den CodeDeploy Agenten.
1.8.1	3. Februar 2026	Behoben: S3-Endpunkt-Bugfix. Hinzugefügt: Sectigo CA-Zertifikat zur Windows-CA-Liste.
1.8.0	31. Juli 2025	Geändert: Das mitgelieferte Ruby wurde im CodeDeploy Agenten für Windows auf 3.2 aktualisiert.
1.7.1	14. November 2024	Geändert: Die Abhängigkeiten für Sicherheitspatches wurden aktualisiert.

Version	Datum der Veröffentlichung	Details
1.7.0	6. März 2024	Hinzugefügt: Eine `:disable_imds_v1:` Konfigurationseinstellung zur CodeDeploy Agenten-Konfigurationsdatei. Verwenden Sie diese Einstellung, um den Fallback auf IMDSv1 zu deaktivieren, wenn IMDSv2-Fehler auftreten. Die Standardeinstellung ist (aktiviert den Fallback). `false` Weitere Informationen finden Sie in der Referenz zur CodeDeploy Agentenkonfiguration. Hinzugefügt: Support für das Betriebssystem Red Hat Enterprise Linux 9 (RHEL 9). Hinzugefügt: Support für die Ruby-Versionen 3.1 und 3.2 auf Ubuntu Server. Behoben: Der CodeDeploy Agent generiert jetzt einen benutzerfreundlichen Fehler, wenn die CodeDeploy Agenten-Konfigurationsdatei nicht geladen werden kann. Geändert: Ruby wurde im CodeDeploy Agenten für Windows auf 2.7.8-1 aktualisiert.
1.6.0	30. März 2023	Hinzugefügt: Support für Ruby 3.1, 3.2. Hinzugefügt: Support für Amazon Linux 2023. Hinzugefügt: Support für Windows Server 2022. Geändert: Die Standardeinstellung von `verbose` ist jetzt `false` für Windows Server-Instanzen. Um weiterhin Debugmeldungen in Protokolldateien unter Windows drucken zu können, müssen Sie `verbose` auf `true` einstellen. Entfernt: Support für Windows Server 2016 und Windows Server 2012 R2. Entfernt: Support für Amazon Linux 2018.03.x.

Version	Datum der Veröffentlichung	Details
1.5.0	03. März 2023	Hinzugefügt: Support für Ruby 3. Hinzugefügt: Support für Ubuntu 22.04. Behoben: Ein Problem, bei dem ein Neustart des CodeDeploy Agenten kurz nach dem Start dazu führte, dass der Agent hängen blieb. Geändert: Der CodeDeploy Agent schlägt jetzt eine Host-Bereitstellung beim Start des Agenten fehl, wenn der Agent-Dienst während der Ausführung eines Hook-Skripts unerwartet neu gestartet wird. Mit diesem Fix können Sie vermeiden, das 70-minütige Timeout abzuwarten, bevor Sie erneut versuchen, eine Bereitstellung durchzuführen. Hinweis zu veralteter Version: CodeDeploy Agent 1.5.0 ist die letzte Version, die Windows Server 2016 und Windows Server 2012 R2 unterstützt. Entfernt: Support für den CodeDeploy Agenten auf Ubuntu 14.04 LTS, Windows Server 2008 R2 und Windows Server 2008 R2 32-Bit.
1.4.1	6. Dezember 2022	Behoben: Sicherheitslücke im Zusammenhang mit der Protokollierung. Verbesserung: Die Protokollierung beim Abfragen des Host-Befehls wurde verbessert.

Version	Datum der Veröffentlichung	Details
1.4.0	31. August 2022	Hinzugefügt: Support für Red Hat Enterprise Linux 8. Hinzugefügt: Support für lange Dateipfade auf dem CodeDeploy Agenten für Windows. Um lange Dateipfade zu aktivieren, müssen Sie den entsprechenden Windows-Registrierungsschlüssel festlegen und dann Ihren Agenten neu starten. Weitere Informationen finden Sie unter Lange Dateipfade führen zu der Fehlermeldung „Keine solche Datei oder kein solches Verzeichnis“. Behoben: Ein Problem beim Entpacken, wenn die Festplatte voll war. Der CodeDeploy Agent erkennt jetzt den Exit-Code 50 des Entpackvorgangs, der auf eine volle Festplatte hinweist, entfernt teilweise entpackte Dateien und löst eine Ausnahme aus, um einen Fehler auf dem CodeDeploy Server zu melden. Die Fehlermeldung ist als Fehlermeldung im Zusammenhang mit einem Lebenszyklusereignis sichtbar, und die Bereitstellung auf Host-Ebene wird beendet, ohne dass es zu einer Unterbrechung kommt oder ein Timeout auftritt. Behoben: Ein Problem, das zum Ausfall des Agenten führen würde. Behoben: Ein Problem, bei dem Hooks während eines außergewöhnlichen Rennens ein Timeout auftraten. Hooks ohne Skripte werden jetzt fortgesetzt und verursachen keine Ausfälle oder Timeouts mehr. Geändert: Das `update` Skript aus dem CodeDeploy `bin` Agentenverzeichnis wurde entfernt, da es nicht mehr verwendet wird. Geändert: Der CodeDeploy Agent für Windows Server bündelt jetzt Ruby 2.7.

Version	Datum der Veröffentlichung	Details
		Geändert: Neue Umgebungsvariablen wurden hinzugefügt, die je nach Quelle des Bereitstellungspakets (Amazon S3 oder GitHub) von Hook-Skripten verwendet werden können. Weitere Informationen finden Sie unter Verfügbarkeit von Umgebungsvariablen für Hooks. **Important** Veraltungshinweis: CodeDeploy Agent 1.4.0 ist die letzte Version, die Installationsprogramme für 32-Bit-Windows Server enthalten wird. Hinweis auf veraltete Versionen: CodeDeploy Agent 1.4.0 ist die letzte Version, die Windows Server 2008 R2 unterstützen wird. Entfernt: Support für den CodeDeploy Agenten auf den folgenden Amazon EC2 EC2-AMIs: Amazon Linux 2014.09, 2016.03, 2016.09 und 2017.03.

Version	Datum der Veröffentlichung	Details
1.3.2	6. Mai 2021	**Important** CodeDeploy Agent 1.3.2-Adressen, die Windows-Hosts betreffen, auf denen der Agent ausgeführt wird. CVE-2018-1000201 Das CVE zitiert ruby-ffi, was eine Abhängigkeit des Agenten ist. CodeDeploy Wenn Ihr Agent mit Amazon EC2 Systems Manager (SSM) installiert wurde und so eingestellt ist, dass er automatisch aktualisiert wird, ist keine Aktion erforderl ich. Andernfalls ist eine Aktion erforderlich, um den Agenten manuell zu aktualisieren. Um den Agenten zu aktualisieren, folgen Sie den Anweisungen unter Den CodeDeploy Agenten auf Windows Server aktualisi eren. Behoben: Ein Problem bei der Installation des CodeDeploy Agenten auf Ubuntu 20.04 und höher. Behoben: Ein zeitweiliges Problem, das beim Extrahieren komprimierter Dateien auftrat, weil relative Pfade nicht korrekt behandelt wurden. Hinzugefügt: Support für AWS PrivateLink und VPC-Endpu nkte für Windows-Instances. Hinzugefügt: AppSpec Dateiverbesserungen, wie unten beschrieben. • Sie können jetzt einen benutzerdefinierten Dateinamen für die AppSpec Datei angeben, wenn Sie eine lokale Bereitste llung erstellen. Weitere Informationen finden Sie unter Erstellen Sie eine lokale Bereitstellung.

Version	Datum der Veröffentlichung	Details
		• Die AppSpec Datei kann jetzt eine `.yaml` Dateierweiterung haben. • Sie können jetzt bereitgestellte Dateien mit einer neuen, optionalen `file_exists_behavior` Einstellung in der AppSpec Datei überschreiben. Weitere Informationen finden Sie unter AppSpec Abschnitt „Dateien“ (nur EC2/On-Premises Bereitstellungen). Aktualisiert: verwendet CodeDeploy jetzt das AWS SDK for Ruby 3.0.
1.3.1	22. Dezember 2020	Behoben: 1.3.0-Problem, das den Start von lokalen Instanzen verhinderte.
1.3.0	10. November 2020	**Important** Diese Version ist veraltet. Behoben: Ein abgelaufenes Zertifikat, das nicht mehr verwendet wurde, wurde entfernt. Problem behoben: Die Eingabeaufforderung wurde aus dem von verwendeten Agenten-Deinstallationsskript entfernt AWS Systems Manager, wodurch es einfacher wurde, einen Host oder eine Flotte auf eine frühere Version des Agenten herunterzustufen.

Version	Datum der Veröffentlichung	Details
1.2.1	23. September 2020	Geändert: Die AWS SDK für Ruby Abhängigkeit wurde von Version 2 auf Version 3 aktualisiert. Hinzugefügt: Support für IMDSv2. Beinhaltet einen stillen Fallback auf IMDSv1, falls IMDSv2-HTTP-Anfragen fehlschlagen. Geändert: Die Abhängigkeiten von Rake und Rubyzip für Sicherheitspatches wurden aktualisiert. Behoben: Stellt sicher, dass eine leere PID-Datei den Status zurückgibt `No CodeDeploy Agent Running` und die PID-Datei beim Start des Agenten bereinigt.

Version	Datum der Veröffentlichung	Details
1.1.2	4. August 2020	Hinzugefügt: Support für Ubuntu Server 19.10 und 20.04. Hinweis: Version 19.10 hat das Ende der Lebensdauer erreicht und wird von Ubuntu oder nicht mehr unterstützt. CodeDeploy Hinzugefügt: Verbesserungen der Speichereffizienz für Linux und Ubuntu, um reservierten Speicher schneller freizugeben. Hinzugefügt: Kompatibilität mit Windows Server „Silent-C leanup“, was dazu führte, dass der Agent in einigen Fällen nicht reagierte. Hinzugefügt: Ignoriere nicht leere Verzeichnisse während der Bereinigung, um Fehler bei der Bereitstellung zu vermeiden. Hinzugefügt: Support für AWS Local Zone in Los Angeles (LA). Hinzugefügt: Extrahieren Sie AZ aus Instanz-Metadaten, um die Kompatibilität für AWS Local Zones zu gewährleisten. Hinzugefügt: Benutzer können ihr Archiv jetzt in Unterverz eichnissen bereitstellen und müssen es nicht mehr im Stammverzeichnis speichern. Hinzugefügt: Es wurde ein Problem mit Rubyzip festgeste llt, das zu Speicherlecks führen könnte. Der Befehl zum Entpacken wurde aktualisiert, sodass zunächst versucht wird, ein vom System installiertes Unzip-Hilfsprogramm zu verwenden, bevor Rubyzip verwendet wird. Hinzugefügt: `:enable_auth_policy:` als Agent-Kon figurationseinstellung.

Version	Datum der Veröffentlichung	Details
		Geändert: Unzip-Warnungen werden jetzt ignoriert, sodass die Bereitstellungen fortgesetzt werden.
1.1.0	30. Juni 2020	Geändert: Die Versionierung des CodeDeploy Agenten folgt jetzt der Ruby-Standard-Versionskonvention. Hinzugefügt: Neuer Parameter für den Befehl „install and update“, um die Installation einer bestimmten Agentenversion über die Befehlszeile zu ermöglichen. Entfernt: Der Auto Updater für den CodeDeploy Agenten für Linux und Ubuntu wurde entfernt. Informationen zur Konfigura tion automatischer Updates des CodeDeploy Agenten finden Sie unter Installieren des CodeDeploy Agenten mit AWS Systems Manager.
1.0.1.1597	15. November 2018	Erweiterung: CodeDeploy unterstützt Ubuntu 18.04. Erweiterung: CodeDeploy unterstützt Ruby 2.5. Verbesserung: CodeDeploy unterstützt FIPS-Endpunkte. Weitere Informationen zu FIPS-Endpunkten finden Sie unter FIPS 140-2-Übersicht. Informationen zu Endpunkten, die mit verwendet werden können CodeBuild, finden Sie unter CodeDeployRegionen und Endpunkte.

Version	Datum der Veröffentlichung	Details
1.0.1.1518	12. Juni 2018	Verbesserung: Es wurde ein Problem behoben, das zu einem Fehler führte, wenn der CodeDeploy Agent geschlossen wurde, während er Umfrageanfragen annimmt. Verbesserung: Es wurde eine Funktion zur Nachverfolgung von Bereitstellungen hinzugefügt, die verhindert, dass der CodeDeploy Agent geschlossen wird, wenn eine Bereitste llung ausgeführt wird. Verbesserung: Verbesserte Leistung beim Löschen von Dateien.
1.0.1.1458	6. März 2018	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Erweiterung: Verbesserte Zertifikatvalidierungen zur Unterstüt zung von mehr vertrauenswürdigen Ausstellern. Verbesserung: Es wurde ein Problem behoben, das dazu führte, dass die lokale CLI während einer Bereitstellung, die ein BeforeInstall Lebenszyklusereignis beinhaltete, fehlschlug. Verbesserung: Es wurde ein Problem behoben, das dazu führen konnte, dass eine aktive Bereitstellung fehlschlug, wenn der CodeDeploy Agent aktualisiert wurde.
1.0.1.1352	16. November 2017	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Feature: Es wurde eine neue Funktion zum Testen und Debuggen einer EC2/On-Premises Bereitstellung auf einem lokalen Computer oder einer Instanz eingeführt, auf der der CodeDeploy Agent installiert ist.

Version	Datum der Veröffentlichung	Details
1.0.1.1106	16. Mai 2017	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Funktion: Einführung neuer Unterstützung für den Umgang mit Inhalten an einem Zielort, der nicht Teil der Anwendung srevision aus der letzten erfolgreichen Bereitstellung war. Bereitstellungsoptionen für vorhandene Inhalte umfassen jetzt das Speichern der Inhalte, das Überschreiben der Inhalte oder das Fehlschlagen der Bereitstellung. Verbesserung: Der CodeDeploy Agent wurde mit Version 2.9.2 von AWS SDK für Ruby (aws-sdk-core 2.9.2) kompatibe l gemacht.
1.0.1.1095	29. März 2017	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Verbesserung: Unterstützung für den CodeDeploy Agenten in der Region China (Peking) eingeführt. Verbesserung: Puppet kann nun auf Windows Server-In stanzen ausgeführt werden, wenn es durch einen Lifecycle-Event-Hook aufgerufen wird. Erweiterung: Verbesserung der Handhabung von `untar`-Vorgängen.

Version	Datum der Veröffentlichung	Details
1.0.1.1067	6. Januar 2017	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Erweiterung: Prüfung zahlreicher Fehlermeldungen zur Einbeziehung spezifischerer Ursachen für Bereitstellungsfehler. Verbesserung: Es wurde ein Problem behoben, durch das der CodeDeploy Agent bei einigen Bereitstellungen nicht die richtige Anwendungsversion für die Bereitstellung identifizieren konnte. Erweiterung: Zurücksetzung der Nutzung von `pushd` und `popd` vor und nach dem `untar`-Vorgang.
1.0.1.1045	21. November 2016	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Verbesserung: Der CodeDeploy Agent wurde mit Version 2.6.11 von AWS SDK für Ruby (aws-sdk-core 2.6.11) kompatibel gemacht.
1.0.1.1037	19. Oktober 2016	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Der CodeDeploy Agent für Amazon Linux-, RHEL- und Ubuntu-Server-Instances wurde mit der folgenden Änderung aktualisiert. Für Windows Server-Instances bleibt die neueste Version 1.0.1.998. Erweiterung: Der Agent kann jetzt bestimmen, welche Version von Ruby auf einer Instance installiert ist, sodass er das `codedeploy-agent` -Skript mit dieser Version aufrufen kann.

Version	Datum der Veröffentlichung	Details
1.0.1.1011.1	17. August 2016	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Erweiterung: Entfernung der Änderungen aus Version 1.0.1.1011 aufgrund von Problemen mit der Shell-Unterstützung. Diese Version des Agenten entspricht funktional Version 1.0.1.998, die am 11. Juli 2016 veröffentlicht wurde.
1.0.1.1011	15. August 2016	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Der CodeDeploy Agent für Amazon Linux-, RHEL- und Ubuntu-Server-Instances wurde mit den folgenden Änderungen aktualisiert. Für Windows Server-Instances bleibt die neueste Version 1.0.1.998. Feature: Unterstützung für das Aufrufen des CodeDeploy Agenten mithilfe der Bash-Shell auf Betriebssystemen hinzugefügt, auf denen das Systemd-Init-System verwendet wird. Verbesserung: Unterstützung für alle Versionen von Ruby 2.x im Agenten und im CodeDeploy Agenten-Updater aktiviert. CodeDeploy Aktualisierte CodeDeploy Agenten sind nicht mehr nur von Ruby 2.0 abhängig. (Ruby 2.0 ist weiterhin für die Deb- und RPM-Versionen des CodeDeploy Agenten-Installationsprogramms erforderlich.)
1.0.1.998	11. Juli 2016	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Verbesserung: Die Unterstützung für die Ausführung des CodeDeploy Agenten mit anderen Benutzerprofilen als Root wurde behoben. Die Variable mit dem Namen `USER` wird durch `CODEDEPLOY_USER` ersetzt, um Konflikte mit Umgebungsvariablen zu verhindern.

Version	Datum der Veröffentlichung	Details
1.0.1.966	16. Juni 2016	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Feature: Unterstützung für die Ausführung des CodeDeploy Agenten mit anderen Benutzerprofilen als Root eingeführt. Erweiterung: Wiederherstellung der Unterstützung für die Angabe der Anzahl an Anwendungsrevisionen, die der CodeDeploy-Agent für eine Bereitstellungsgruppe archivieren soll. Verbesserung: Der CodeDeploy Agent wurde mit Version 2.3 von AWS SDK für Ruby (aws-sdk-core 2.3) kompatibel gemacht. Verbesserung: Probleme mit UTF-8 der Kodierung bei Bereitstellungen wurden behoben. Erweiterung: Verbesserung der Genauigkeit bei der Identifik ation von Prozessnamen.
1.0.1.950	24. März 2016	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Funktion: Hinzufügung von Proxy-Support für die Installation. Verbesserung: Das Installationsskript wurde aktualisiert, sodass der CodeDeploy Agent nicht heruntergeladen wird, wenn die neueste Version bereits installiert ist.
1.0.1.934	11. Februar 2016	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Funktion: Einführung der Unterstützung für die Angabe der Anzahl an Anwendungsrevisionen, die der CodeDeploy-Agent für eine Bereitstellungsgruppe archivieren soll.

Version	Datum der Veröffentlichung	Details
1.0.1.880	11. Januar 2016	Hinweis: Diese Version wird nicht mehr unterstützt und kann zum Fehlschlagen von Bereitstellungen führen. Verbesserung: Der CodeDeploy Agent wurde mit Version 2.2 von AWS SDK für Ruby (aws-sdk-core 2.2) kompatibel gemacht. Version 2.1.2 wird weiterhin unterstützt.
1.0.1.854	17. November 2015	Hinweis: Diese Version wird nicht mehr unterstützt. Wenn Sie diese Version verwenden, schlagen Ihre Bereitstellungen fehl. Feature: Unterstützung für den Hash-Algorithmus eingeführt. SHA-256 Funktion: Einführung von Unterstützung für die Versionsk ontrolle in `.version`-Dateien. Funktion: Herstellung der Verfügbarkeit der Bereitstellungsgru ppen-ID durch Verwendung einer Umgebungsvariablen. Verbesserung: Unterstützung für die Überwachung von CodeDeploy Agentenprotokollen mithilfe von Amazon CloudWatch Logs hinzugefügt.

Zugehörige Informationen finden Sie unter den folgenden Themen:

- Ermitteln Sie die Version des Agenten CodeDeploy
- Installieren Sie den CodeDeploy Agenten

Eine Historie der CodeDeploy Agentenversionen finden Sie im Release-Repository unter GitHub.

Den CodeDeploy Prozess verwalten

Der CodeDeploy Agent verwendet Systemd, um den Agentenprozess auf allen unterstützten Linux-Distributionen zu verwalten.

Für Version 2.0.x und höher liefert der Agent nur eine native Systemd-Unit () aus. `codedeploy-agent.service` Es gibt kein Skript. `/etc/init.d/codedeploy-agent` Wird verwendet, `systemctl` um den Agenten zu verwalten oder die Binärdatei direkt mit `status` Unterbefehlen`start`, `stoprestart`, oder aufzurufen.

Für Version 1.8.x und früher werden sowohl die RPM- als auch die Deb-Distribution mit Startskripten unter ausgeliefert. `/etc/init.d/codedeploy-agent` Auf Systemen, die Systemd unterstützen, kann die Verwendung des `service` Befehls versehentlich die init.d-Skripten ausführen, anstatt Systemd die Verwaltung des Prozesses zu ermöglichen. Wird für Agentenoperationen verwendet, um dieses `systemctl` Problem zu vermeiden.

Verwenden Sie das folgende `systemctl` Hilfsprogramm, um den CodeDeploy Agenten zu verwalten:

```
sudo systemctl start codedeploy-agent
sudo systemctl stop codedeploy-agent
sudo systemctl restart codedeploy-agent
sudo systemctl status codedeploy-agent
```

Unter macOS läuft der Agent als launchd-Daemon`com.amazon.aws.codedeploy-agent`, definiert in`/Library/LaunchDaemons/com.amazon.aws.codedeploy-agent.plist`. launchd startet den Agenten beim Systemstart und startet ihn neu, wenn er abstürzt. Verwenden Sie das folgende `launchctl` Hilfsprogramm, um den Agenten auf macOS zu verwalten:

```
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
```

Diese Befehle starten, stoppen, neu starten und zeigen jeweils den Status des Agenten an.

Revision der Anwendung und Bereinigung der Protokolldateien

Der CodeDeploy Agent archiviert Revisionen und Protokolldateien auf Instanzen. Der CodeDeploy Agent bereinigt diese Artefakte, um Speicherplatz zu sparen.

Protokolle zur Bereitstellung von Anwendungsrevisionen: Sie können die Option:max_revisions: in der Agenten-Konfigurationsdatei verwenden, um die Anzahl der zu archivierenden

Anwendungsrevisionen durch Eingabe einer beliebigen positiven Ganzzahl anzugeben. CodeDeploy archiviert auch die Protokolldateien für diese Revisionen. Alle anderen werden gelöscht, mit Ausnahme der Protokolldatei der letzten erfolgreichen Bereitstellung. Diese Protokolldatei wird immer aufbewahrt, auch wenn die Anzahl der fehlgeschlagenen Bereitstellungen die Anzahl der aufbewahrten Revisionen überschreitet. Wenn kein Wert angegeben ist, werden die fünf neuesten Versionen zusätzlich zur aktuell bereitgestellten Version CodeDeploy beibehalten.

CodeDeploy Logs: Bei Amazon Linux-, Ubuntu Server-, RHEL- und macOS-Instances rotiert der CodeDeploy Agent die Protokolldateien im `/var/log/aws/codedeploy-agent` Ordner. Die Protokolldatei wird täglich um 00:00:00 Uhr (Instanzzeit) rotiert, oder wenn die Datei 64 MB erreicht, je nachdem, was zuerst eintritt. Wenn am selben Tag eine zweite Rotation stattfindet, hängt der Agent ein numerisches Suffix an (z. B.). `codedeploy-agent.20260715.1.log` Der Agent löscht die Protokolldateien nach sieben Tagen. Das Benennungsmuster für rotierte Protokolldateien ist `codedeploy-agent.`*`YYYYMMDD`*`.log`.

Vom Agenten installierte Dateien CodeDeploy

Der CodeDeploy Agent speichert Revisionen, den Bereitstellungsverlauf und Bereitstellungsskripts in seinem Stammverzeichnis auf einer Instanz. Der Standardname und Speicherort für dieses Verzeichnis lauten:

`'/opt/codedeploy-agent/deployment-root'`für Amazon Linux-, Ubuntu Server-, RHEL- und macOS-Instances.

`'C:\ProgramData\Amazon\CodeDeploy'`für Windows Server-Instanzen.

Sie können die Einstellung root_dir in der CodeDeploy Agent-Konfigurationsdatei verwenden, um den Namen und den Speicherort des Verzeichnisses zu konfigurieren. Weitere Informationen finden Sie unter CodeDeploy Referenz zur Agentenkonfiguration.

Es folgt ein Beispiel für die Datei- und Verzeichnisstruktur unter dem Stammverzeichnis. Die Struktur geht davon aus, dass es N Bereitstellungsgruppen gibt und jede Bereitstellungsgruppe N Bereitstellungen enthält.

```
|--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
```

- Deployment Group ID-Ordner stehen für Ihre einzelnen Bereitstellungsgruppen. Der Verzeichnisname einer Bereitstellungsgruppe ist ihre ID (z. B. `acde1916-9099-7caf-fd21-012345abcdef`). Jedes Verzeichnis einer Bereitstellungsgruppe enthält ein Unterverzeichnis für jede versuchte Bereitstellung in der jeweiligen Gruppe.

  Sie können den Befehl batch-get-deployments verwenden, um eine Bereitstellungsgruppen-ID zu finden.

- Deployment ID-Ordner stehen für die einzelnen Bereitstellungen in einer Bereitstellungsgruppe. Jeder Verzeichnisname einer Bereitstellung ist ihre ID. Jeder Ordner enthält:
  - bundle.tar, eine komprimierte Datei mit dem Inhalt der Bereitstellungsrevision. Verwenden Sie ein ZIP-Hilfsprogramm zum Dekomprimieren, wenn Sie die Revision anzeigen möchten.
  - deployment-archive, ein Verzeichnis mit den Inhalten der Bereitstellungsrevision.
  - logs, ein Verzeichnis mit einer `scripts.log`-Datei. In dieser Datei wird die Ausgabe aller in der Bereitstellungsdatei angegebenen Skripts aufgeführt. AppSpec

  Wenn Sie den Ordner für eine Bereitstellung suchen möchten, aber dessen Bereitstellungs- oder Bereitstellungsgruppen-ID nicht kennen, können Sie die AWS CodeDeploy Konsole oder die verwenden, AWS CLI um sie zu finden. Weitere Informationen finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen.

  Die standardmäßige maximale Anzahl von Bereitstellungen, die in einer Bereitstellungsgruppe archiviert werden können, ist fünf. Wenn diese Anzahl erreicht ist, werden künftige Bereitstellungen archiviert und das älteste Archiv gelöscht. Sie können die Einstellung max_revisions in der CodeDeploy Agentenkonfigurationsdatei verwenden, um die Standardeinstellung zu ändern. Weitere Informationen finden Sie unter CodeDeploy Referenz zur Agentenkonfiguration.

  > **Note**
  >
  > Wenn Sie den von archivierten Bereitstellungen belegten Festplattenspeicher zurückgewinnen möchten, aktualisieren Sie die Einstellung max_revisions auf eine niedrige Zahl, etwa 1 oder 2. Bei der nächsten Bereitstellung werden archivierte Bereitstellungen gelöscht, sodass die Anzahl der von Ihnen angegebenen Anzahl entspricht.

- deployment-instructions enthält vier Textdateien für jede Bereitstellungsgruppe:
  - [Deployment Group ID] -cleanup, eine Textdatei mit einer Undo-Version jedes Befehls, der während einer Bereitstellung ausgeführt wird. Der Dateiname lautet z. B. `acde1916-9099-7caf-fd21-012345abcdef-cleanup`.

- [Deployment Group ID]-install.json, eine JSON-Datei, die während der letzten Bereitstellung erstellt wurde. Sie enthält die Befehle, die während der Bereitstellung ausgeführt werden. Der Dateiname lautet z. B. `acde1916-9099-7caf-fd21-012345abcdef-install.json`.
- [Deployment Group ID] _last_successful_install, eine Textdatei, die das Archivverzeichnis der letzten erfolgreichen Bereitstellung auflistet. Diese Datei wird erstellt, wenn der CodeDeploy Agent alle Dateien in der Bereitstellungsanwendung auf die Instanz kopiert hat. Sie wird vom CodeDeploy Agenten bei der nächsten Bereitstellung verwendet, um zu bestimmen, welche `ApplicationStop` `BeforeInstall` Skripte ausgeführt werden sollen. Der Dateiname lautet z. B. `acde1916-9099-7caf-fd21-012345abcdef_last_successful_install`.
- [Deployment Group ID]_most_recent_install, eine Textdatei, die den Namen des Archivverzeichnisses der letzten Bereitstellung auflistet. Diese Datei wird erstellt, wenn die Dateien in der Bereitstellung erfolgreich heruntergeladen wurden. Die Datei [Deployment Group ID] _last_successful_install wird nach dieser Datei erstellt, wenn die heruntergeladenen Dateien an ihren endgültigen Speicherort kopiert werden. Der Dateiname lautet z. B. `acde1916-9099-7caf-fd21-012345abcdef_most_recent_install`.

- deployment-logs enthält die folgenden Protokolldateien:
  - codedeploy-agent.yyyymmdd.log-Dateien werden für jeden Tag erstellt, an dem eine Bereitstellung erfolgt. Jede Protokolldatei enthält Informationen zu den Bereitstellungen des jeweiligen Tages. Diese Protokolldateien können nützlich sein, um Probleme wie ein Berechtigungsproblem zu beheben. Diese Protokolldatei heißt zunächst `codedeploy-agent.log`. Am nächsten Tag wird das Datum der Bereitstellung in den Dateinamen eingefügt. Ist das Tagesdatum beispielsweise der 3. Januar 2018, können Sie Informationen zu allen Bereitstellungen in `codedeploy-agent.log` finden. Morgen, am 4. Januar 2018, wird die Protokolldatei nach `codedeploy-agent.20180103.log` umbenannt.
  - codedeploy-agent-deployments.log kompiliert den Inhalt von `scripts.log`-Dateien für jede Bereitstellung. Die `scripts.log`-Dateien befinden sich im `logs`-Unterordner unter dem jeweiligen `Deployment ID`-Ordner. Den Einträgen in dieser Datei ist eine Deployment-ID vorangestellt. Beispiel: `"[d-ABCDEF123]LifecycleEvent - BeforeInstall"` kann während einer Bereitstellung mit der ID `d-ABCDEF123` erscheinen. Wenn `codedeploy-agent-deployments.log` die maximale Größe erreicht ist, schreibt der CodeDeploy Agent weiter in die Datei und löscht gleichzeitig alte Inhalte.

Verwaltung des CodeDeploy Agentenbetriebs

Die Anweisungen in diesem Abschnitt zeigen Ihnen, wie Sie den CodeDeploy Agenten installieren, deinstallieren, neu installieren oder aktualisieren und wie Sie überprüfen können, ob der CodeDeploy Agent ausgeführt wird.

> **Tip**
>
> Einige Anweisungen auf dieser Seite unterscheiden sich je nach Agent-Version. Wenn Sie sich nicht sicher sind, welche Version installiert ist, führen Sie `/opt/codedeploy-agent/bin/codedeploy-agent --version` (Linux oder macOS) oder `& 'C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe' --version` (Windows) aus. Wenn dieser Befehl nicht erkannt wird, verwenden Sie Version 1.8.x oder früher. Weitere Informationen finden Sie unter Ermitteln Sie die Version des Agenten CodeDeploy.

Themen

- Stellen Sie sicher, dass der CodeDeploy Agent läuft
- Ermitteln Sie die Version des Agenten CodeDeploy
- Installieren Sie den CodeDeploy Agenten
- Aktualisieren Sie den CodeDeploy Agenten
- Deinstallieren Sie den CodeDeploy Agenten
- CodeDeploy Agentenprotokolle senden an CloudWatch

Stellen Sie sicher, dass der CodeDeploy Agent läuft

In diesem Abschnitt werden Befehle beschrieben, die ausgeführt werden müssen, wenn Sie vermuten, dass der CodeDeploy Agent auf einer Instance nicht mehr ausgeführt wird.

Themen

- Stellen Sie sicher, dass der CodeDeploy Agent für Amazon Linux oder RHEL läuft
- Stellen Sie sicher, dass der CodeDeploy Agent für Ubuntu Server läuft
- Stellen Sie sicher, dass der CodeDeploy Agent für Windows Server läuft
- Stellen Sie sicher, dass der CodeDeploy Agent für macOS läuft

Stellen Sie sicher, dass der CodeDeploy Agent für Amazon Linux oder RHEL läuft

Um zu sehen, ob der CodeDeploy Agent installiert ist und läuft, melden Sie sich bei der Instance an und führen Sie den folgenden Befehl aus:

```
systemctl status codedeploy-agent
```

Wenn der Befehl einen Fehler zurückgibt, ist der CodeDeploy Agent nicht installiert. Installieren Sie ihn, wie in Installieren Sie den CodeDeploy Agenten für Amazon Linux oder RHEL beschrieben.

Wenn der CodeDeploy Agent installiert ist und läuft, sollte eine Meldung wie `The AWS CodeDeploy agent is running.` (Exit-Code 0) angezeigt werden.

Wenn Sie eine Meldung wie `The AWS CodeDeploy agent is not running.` (Exit-Code 3) sehen, starten Sie den Dienst und führen Sie nacheinander die folgenden beiden Befehle aus:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Für Version 2.0.x und höher gibt der `codedeploy-agent status` Befehl die LSB-compatible Exit-Codes zurück: `0` (läuft), `3` (läuft nicht), `4` (unbekannt). Bei der Verwendung `systemctl status` wird der Status der Einheit angezeigt (z. B.`Active: active (running)`).

Stellen Sie sicher, dass der CodeDeploy Agent für Ubuntu Server läuft

Um zu überprüfen, ob der CodeDeploy Agent installiert ist und läuft, melden Sie sich bei der Instanz an und führen Sie den folgenden Befehl aus:

```
systemctl status codedeploy-agent
```

Wenn der Befehl einen Fehler zurückgibt, ist der CodeDeploy Agent nicht installiert. Installieren Sie ihn, wie in Installieren Sie den CodeDeploy Agenten für Ubuntu Server beschrieben.

Wenn der CodeDeploy Agent installiert ist und läuft, sollte eine Meldung wie `The AWS CodeDeploy agent is running.` (Exit-Code 0) angezeigt werden.

Wenn Sie eine Meldung wie `The AWS CodeDeploy agent is not running.` (Exit-Code 3) sehen, starten Sie den Dienst und führen Sie nacheinander die folgenden beiden Befehle aus:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Stellen Sie sicher, dass der CodeDeploy Agent für Windows Server läuft

Um zu überprüfen, ob der CodeDeploy Agent installiert ist und läuft, melden Sie sich bei der Instanz an und führen Sie den folgenden Befehl aus:

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
Status   Name               DisplayName
------   ----               -----------
Running codedeployagent    CodeDeploy Host Agent Service
```

Wenn der Befehl einen Fehler zurückgibt, ist der CodeDeploy Agent nicht installiert. Installieren Sie ihn, wie in Installieren Sie den CodeDeploy Agenten für Windows Server beschrieben.

Wenn für `Status` etwas anderes als `Running` angezeigt wird, starten Sie den Service mit dem folgenden Befehl:

```
powershell.exe -Command Start-Service -Name codedeployagent
```

Sie können den Service mit dem folgenden Befehl neu starten:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Sie können den Service mit dem folgenden Befehl stoppen:

```
powershell.exe -Command Stop-Service -Name codedeployagent
```

Stellen Sie sicher, dass der CodeDeploy Agent für macOS läuft

Um zu überprüfen, ob der CodeDeploy Agent installiert ist und läuft, melden Sie sich bei der Instanz an und führen Sie den folgenden Befehl aus:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Wenn der Befehl einen Fehler zurückgibt, weil die Datei nicht gefunden wurde, ist der CodeDeploy Agent nicht installiert. Installieren Sie ihn, wie in Installieren Sie den CodeDeploy Agenten für macOS beschrieben.

Wenn der CodeDeploy Agent installiert ist und läuft, sollte eine Meldung wie `The AWS CodeDeploy agent is running.` (Exit-Code 0) angezeigt werden.

Wenn Sie eine Meldung wie `The AWS CodeDeploy agent is not running.` (Exit-Code 3) sehen, starten Sie den Agenten mit dem folgenden Befehl:

```
sudo launchctl kickstart system/com.amazon.aws.codedeploy-agent
```

Wenn dieser Befehl meldet, dass der Dienst nicht gefunden werden konnte, wird der launchd-Daemon nicht geladen. Laden und starten Sie ihn mit dem folgenden Befehl:

```
sudo launchctl bootstrap system /Library/LaunchDaemons/com.amazon.aws.codedeploy-
agent.plist
```

Um den Status des gestarteten Daemons zu sehen, führen Sie folgenden Befehl aus:

```
sudo launchctl print system/com.amazon.aws.codedeploy-agent
```

Ermitteln Sie die Version des Agenten CodeDeploy

Sie können die Version des CodeDeploy Agenten, der auf Ihrer Instance ausgeführt wird, auf zwei Arten ermitteln.

Verwenden Sie zunächst für Version 2.0.x und höher den `codedeploy-agent --version` Befehl auf allen Plattformen.

Zweitens können Sie die Version in einer `.version` Datei auf der Instanz anzeigen. Auf Amazon Linux, RHEL, Ubuntu Server und macOS befindet sich die Datei unter `/opt/codedeploy-agent/.version` und auf Windows Server unter`C:\ProgramData\Amazon\CodeDeploy\.version`. Der Inhalt der Datei unterscheidet sich je nach Version:

- Für Version 2.0.x und höher enthält die Datei die bloße Versionsnummer im Format`agent_version: 2.0.0`. Dieser Wert entspricht der Version, die der Agent dem CodeDeploy Dienst meldet.
- Für Version 1.8.x und früher verwendet die Versionszeichenfolge das Format `OFFICIAL_1.0.1.854_rpm` (oder`_deb`,`_msi`).

Themen

- Ermitteln Sie die Version auf Amazon Linux oder RHEL
- Ermitteln Sie die Version auf Ubuntu Server
- Ermitteln Sie die Version auf Windows Server
- Ermitteln Sie die Version auf macOS
- Ermitteln Sie die neueste verfügbare Agentenversion in einem AWS-Region

Ermitteln Sie die Version auf Amazon Linux oder RHEL

Melden Sie sich bei der Instance an, und führen Sie den folgenden Befehl aus:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Für Versionen vor 2.0.0 können Sie auch Folgendes verwenden:

```
sudo yum info codedeploy-agent
```

Ermitteln Sie die Version auf Ubuntu Server

Melden Sie sich bei der Instance an, und führen Sie den folgenden Befehl aus:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Für Versionen vor 2.0.0 können Sie auch Folgendes verwenden:

```
sudo dpkg -s codedeploy-agent
```

Ermitteln Sie die Version auf Windows Server

Melden Sie sich bei der Instance an, und führen Sie den folgenden Befehl aus:

```
& 'C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe' --version
```

Ermitteln Sie die Version auf macOS

Melden Sie sich bei der Instance an, und führen Sie den folgenden Befehl aus:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Sie können auch die Version des installierten Pakets sehen:

```
pkgutil --pkg-info com.amazon.aws.codedeploy-agent
```

Ermitteln Sie die neueste verfügbare Agentenversion in einem AWS-Region

Um herauszufinden, welche Version in der Region, in der Sie die Bereitstellung durchführen, die neueste Version ist, fragen AWS-Region Sie den CodeDeploy Resource Kit-Bucket für diese Region ab. Jeder regionale Bucket enthält eine `latestv2/LATEST_VERSION` Datei mit den Namen der Agentenpakete, die derzeit in dieser Region veröffentlicht wurden. Die Datei ist öffentlich lesbar, sodass Sie keine Anmeldeinformationen benötigen, um sie abzurufen.

```
curl https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/LATEST_VERSION
```

Ersetzen Sie *`bucket-name`* und *`region-identifier`* durch den Resource Kit-Bucket und die Region-ID für die Region, in der Sie die Bereitstellung durchführen. Die Liste finden Sie unter Bucket-Namen für das Resource Kit nach Regionen.

Daraufhin erhalten Sie ein Ergebnis, das dem hier dargestellten entspricht:

```
{
    "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"
}
```

Jeder Schlüssel benennt einen Pakettyp und für RPM und DEB auch die Instanzarchitektur. Jeder Wert ist der Amazon S3 S3-Objektschlüssel dieses Pakets, und die Agentenversion und Build-

Nummer sind Teil seines Dateinamens im Format*agent-version*-*build-number*. In der vorherigen Ausgabe ist die neueste Version, die in der Region verfügbar ist, 2.1.0.

Für Version 2.1.2 und höher hat die Datei auch einen `pkg_aarch64` Schlüssel, der das macOS-Installationspaket benennt.

Informationen zur Installation oder Aktualisierung der Version, die diese Datei benennt, finden Sie unter Installieren Sie den CodeDeploy Agenten undAktualisieren Sie den CodeDeploy Agenten. Informationen zu den in den einzelnen Versionen enthaltenen Funktionen finden Sie unterVersionsverlauf des Agenten CodeDeploy.

Installieren Sie den CodeDeploy Agenten

Für die Verwendung CodeDeploy auf EC2-Instances oder lokalen Servern muss der CodeDeploy Agent zuerst installiert werden. Sie können den CodeDeploy Agenten mit AWS Systems Manager oder über die Befehlszeile installieren. Weitere Informationen zu Systems Manager finden Sie unter Was ist AWS Systems Manager.

Die empfohlene Version für die Installation finden Sie unterVersionsverlauf des Agenten CodeDeploy.

Themen

- Installieren Sie den CodeDeploy Agenten mit AWS Systems Manager
- Installieren Sie den CodeDeploy Agenten über die Befehlszeile

Installieren Sie den CodeDeploy Agenten mit AWS Systems Manager

Sie können das AWS-Managementkonsole oder das verwenden AWS CLI , um den CodeDeploy Agenten auf Ihren Amazon EC2- oder lokalen Instances zu installieren, indem Sie AWS Systems Manager Sie können wählen, ob Sie eine bestimmte Version installieren oder immer die neueste Version des Agenten installieren möchten. Weitere Informationen zu finden Sie AWS Systems Manager unter Was ist. AWS Systems Manager

Sie können den CodeDeploy Agenten auch von einem Amazon S3 S3-Bucket aus installieren. Informationen zum Verwenden eines Amazon-S3-Download-Links finden Sie unter Installieren Sie den CodeDeploy Agenten über die Befehlszeile.

Themen

- Voraussetzungen
- Installieren Sie den CodeDeploy Agenten

Voraussetzungen

Folgen Sie den Schritten unterErste Schritte mit CodeDeploy, um IAM-Berechtigungen einzurichten und die AWS CLI.

Wenn Sie den CodeDeploy Agenten auf einem lokalen Server mit Systems Manager installieren, müssen Sie Ihren lokalen Server bei Amazon EC2 Systems Manager registrieren. Weitere Informationen finden Sie im AWS Systems Manager Benutzerhandbuch unter Systems Manager in Hybridumgebungen einrichten.

Installieren Sie den CodeDeploy Agenten

Bevor Sie Systems Manager zur Installation des CodeDeploy Agenten verwenden können, müssen Sie sicherstellen, dass die Instanz korrekt für Systems Manager konfiguriert ist.

Installation oder Aktualisierung des SSM-Agenten

Auf einer Amazon EC2 EC2-Instance benötigt der CodeDeploy Agent, dass auf der Instance Version 2.3.274.0 oder höher ausgeführt wird. Bevor Sie den CodeDeploy Agenten installieren, aktualisieren oder installieren Sie den SSM-Agent auf der Instance, falls Sie dies noch nicht getan haben.

Der SSM-Agent ist auf einigen Amazon EC2 EC2-AMIs vorinstalliert, die von bereitgestellt werden. AWS Weitere Informationen finden Sie unter Amazon Machine Images (AMIs) mit vorinstalliertem SSM-Agent.

> **Note**
>
> Stellen Sie sicher, dass das Betriebssystem der Instance auch vom Agenten unterstützt wird. CodeDeploy Weitere Informationen finden Sie unter Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy.

Informationen zur Installation oder Aktualisierung des SSM-Agenten auf einer Instanz, auf der Linux ausgeführt wird, finden Sie im AWS Systems Manager Benutzerhandbuch unter Installation und Konfiguration des SSM-Agenten auf Linux-Instances.

Informationen zur Installation oder Aktualisierung des SSM-Agenten auf einer Instanz, auf der Windows Server ausgeführt wird, finden Sie im Benutzerhandbuch unter Installation und Konfiguration des SSM-Agenten auf Windows-Instanzen.AWS Systems Manager

Informationen zur Installation oder Aktualisierung des SSM-Agenten auf einer EC2-Mac-Instanz finden Sie unter Manuelles Installieren und Deinstallieren des SSM-Agenten auf EC2-Instances für macOS im Benutzerhandbuch.AWS Systems Manager

(Optional) Überprüfen der Voraussetzungen für Systems Manager

Bevor Sie Systems Manager Run Command zur Installation des CodeDeploy Agenten verwenden, stellen Sie sicher, dass Ihre Instances die Mindestanforderungen von Systems Manager erfüllen. Weitere Informationen finden Sie unter Einrichten von AWS Systems Manager im Benutzerhandbuch für AWS Systems Manager .

Installieren Sie den CodeDeploy Agenten

Mit SSM können Sie den CodeDeploy einmal installieren oder einen Zeitplan für die Installation neuer Versionen einrichten.

Um den CodeDeploy Agenten zu installieren, wählen Sie das Systems Manager Distributor-Paket für die gewünschte Agentenversion aus, und folgen Sie dann den Schritten unter Pakete mit dem AWS Systems Manager Distributor installieren oder aktualisieren. Das Paket unterscheidet sich je nach Agentenversion:

- Wählen Sie für Version 2.0.x und höher das `AWSCodeDeployAgentV2` Paket aus. Dies ist das einzige Paket, das macOS ab Version 2.1.2 unterstützt.
- Wählen Sie für Version 1.8.x und früher das Paket aus. `AWSCodeDeployAgent`

Installieren Sie den CodeDeploy Agenten über die Befehlszeile

Verwenden Sie die folgenden Themen, um den CodeDeploy Agenten über die Befehlszeile zu installieren und auszuführen.

Themen

- Installieren Sie den CodeDeploy Agenten für Amazon Linux oder RHEL
- Installieren Sie den CodeDeploy Agenten für Ubuntu Server
- Installieren Sie den CodeDeploy Agenten für Windows Server
- Installieren Sie den CodeDeploy Agenten für macOS

Installieren Sie den CodeDeploy Agenten für Amazon Linux oder RHEL

Melden Sie sich bei der Instance an, und führen Sie die folgenden Befehle nacheinander aus. Es hat sich bewährt, den Befehl `sudo yum update` zuerst auszuführen, wenn es `yum` um die Installation von Paketen geht. Sie können ihn jedoch überspringen, wenn Sie nicht alle Ihre Pakete aktualisieren möchten.

```
sudo yum update
```

Für Version 2.0.x und höher:

```
sudo yum install wget
```

Für Version 1.8.x und früher:

```
sudo yum install ruby wget
```

(Optional) Führen Sie das folgende Skript aus, um das AMI von allen vorherigen Agent-Caching-Informationen zu säubern:

```
#!/bin/bash
CODEDEPLOY_BIN="/opt/codedeploy-agent/bin/codedeploy-agent"
$CODEDEPLOY_BIN stop
yum erase codedeploy-agent -y
```

Wechseln Sie zu Ihrem Home-Verzeichnis:

```
cd /home/ec2-user
```

> **Note**
>
> `/home/ec2-user`Stellt im vorherigen Befehl den Standardbenutzernamen für eine Amazon Linux- oder RHEL Amazon EC2 EC2-Instance dar. Wenn Ihre Instance mit einer benutzerdefinierten AMI erstellt wurde, kann der Eigentümer der AMI einen anderen Standardbenutzernamen angegeben haben.

Laden Sie das CodeDeploy Agent-Installationsprogramm herunter. Das `latestv2/` Präfix dient dem Installationsprogramm der Version 2.0.x und das `latest/` Präfix dient dem Installationsprogramm der Version 1.8.x.

Für Version 2.0.x und höher:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/install
```

Für Version 1.8.x und früher:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```

*bucket-name*ist der Name des Amazon S3 S3-Buckets, der die CodeDeploy Resource Kit-Dateien für Ihre Region enthält, und *region-identifier* ist die Kennung für Ihre Region.

Zum Beispiel für Version 2.0.x und höher:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/install`

Für Version 1.8.x und früher:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install`

Eine Liste der Bucket-Namen und Regionsbezeichner finden Sie unter. Bucket-Namen für das Resource Kit nach Regionen

Legen Sie Ausführungsberechtigungen für die `install` Datei fest:

```
chmod +x ./install
```

Gehen Sie wie folgt vor, um die neueste Version des CodeDeploy Agenten zu installieren:

-
```
sudo ./install auto
```

Gehen Sie wie folgt vor, um eine bestimmte Version des CodeDeploy Agenten zu installieren:

- Listet die verfügbaren Versionen in Ihrer Region auf:

```
aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier
 | grep '\.rpm$'
```

- Installieren Sie eine der folgenden Versionen:

  Für Version 2.0.x und höher sind die Pakete architekturspezifisch:

  ```
  sudo ./install auto -v releases/codedeploy-agent-version.arch.rpm
  ```

  Wo *arch* ist `x86_64` oder `aarch64` hängt von Ihrer Instance-Architektur ab.

  Für Version 1.8.x und früher verwenden Pakete: `noarch`

  ```
  sudo ./install auto -v releases/codedeploy-agent-version.noarch.rpm
  ```

  Die neueste veröffentlichte Version finden Sie unter. Versionsverlauf des Agenten CodeDeploy

Führen Sie den folgenden Befehl aus, um zu prüfen, ob der Dienst ausgeführt wird:

```
systemctl status codedeploy-agent
```

Für Version 2.0.x und höher sollte, wenn der Agent ausgeführt wird, eine Meldung wie `The AWS CodeDeploy agent is running.`

Wenn der Agent für Version 1.8.x und früher läuft, sollten Sie eine Meldung wie die folgende sehen `The AWS CodeDeploy agent is running.`

Wenn der Agent nicht läuft, starten Sie den Dienst und überprüfen Sie den Status:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Installieren Sie den CodeDeploy Agenten für Ubuntu Server

Um den CodeDeploy Agenten auf Ubuntu Server zu installieren

1. Melden Sie sich bei der Instanz an.

2. Geben Sie nacheinander die folgenden Befehle ein:

```
sudo apt update
```

Für Version 2.0.x und höher:

```
sudo apt install wget
```

Für Version 1.8.x und früher:

```
sudo apt install ruby-full wget
```

3. Geben Sie den folgenden Befehl ein:

```
cd /home/ubuntu
```

*/home/ubuntu*steht für den Standardbenutzernamen für eine Ubuntu-Server-Instanz. Wenn Ihre Instance mit einer benutzerdefinierten AMI erstellt wurde, kann der Eigentümer der AMI einen anderen Standardbenutzernamen angegeben haben.

4. Geben Sie den folgenden Befehl ein. Das `latestv2/` Präfix dient dem Installationsprogramm der Version 2.0.x und das `latest/` Präfix dient dem Installationsprogramm der Version 1.8.x.

Für Version 2.0.x und höher:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/install
```

Für Version 1.8.x und früher:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```

*bucket-name*ist der Name des Amazon S3 S3-Buckets, der die CodeDeploy Resource Kit-Dateien für Ihre Region enthält, und *region-identifier* ist die Kennung für Ihre Region.

Zum Beispiel für Version 2.0.x und höher:

```
https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/
install
```

Für Version 1.8.x und früher:

```
https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/
install
```

Eine Liste der Bucket-Namen und Regionsbezeichner finden Sie unter. Bucket-Namen für das Resource Kit nach Regionen

5. Geben Sie den folgenden Befehl ein:

```
chmod +x ./install
```

6. Führen Sie eine der folgenden Aktionen aus:

   - Gehen Sie wie folgt vor, um die neueste Version des CodeDeploy Agenten zu installieren:

     ```
     sudo ./install auto
     ```

   - Gehen Sie wie folgt vor, um eine bestimmte Version des CodeDeploy Agenten zu installieren:

     - Listet die verfügbaren Versionen in Ihrer Region auf:

       ```
       aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-
       identifier | grep '\.deb$'
       ```

     - Installieren Sie eine der folgenden Versionen:

       Für Version 2.0.x und höher sind die Pakete architekturspezifisch:

       ```
       sudo ./install auto -v releases/codedeploy-agent_version_arch.deb
       ```

       Wo *arch* ist amd64 oder arm64 hängt von Ihrer Instance-Architektur ab.

       Für Version 1.8.x und früher verwenden Pakete eine andere Namenskonvention:

       ```
       sudo ./install auto -v releases/codedeploy-agent_version_all.deb
       ```

       Die neueste veröffentlichte Version finden Sie unter. Versionsverlauf des Agenten CodeDeploy

Um zu überprüfen, ob der Dienst ausgeführt wird

1. Geben Sie den folgenden Befehl ein:

   ```
   systemctl status codedeploy-agent
   ```

   Wenn der CodeDeploy Agent installiert ist und läuft:

   Für Version 2.0.x und höher sollten Sie Folgendes sehen: `The AWS CodeDeploy agent is running.`

   Für Version 1.8.x und früher sollten Sie Folgendes sehen: `The AWS CodeDeploy agent is running.`
2. Wenn der Agent nicht läuft, starten Sie den Dienst und überprüfen Sie den Status:

   ```
   systemctl start codedeploy-agent
   ```

   ```
   systemctl status codedeploy-agent
   ```

Installieren Sie den CodeDeploy Agenten für Windows Server

Auf Windows Server-Instanzen können Sie eine der folgenden Methoden verwenden, um den CodeDeploy Agenten herunterzuladen und zu installieren:

- Verwenden AWS Systems Manager (empfohlen)
- Führen Sie eine Reihe von PowerShell Windows-Befehlen aus.
- Wählen eines direkten Download-Links
- Führen Sie einen Amazon S3 S3-Kopierbefehl aus.

> **Note**
>
> Der Ordner, in dem der CodeDeploy Agent installiert ist, ist`C:\ProgramData\Amazon \CodeDeploy`. Platzieren Sie in diesem Pfad keine Verzeichnisverbindungen oder symbolischen Links. Für Version 2.0.x und höher setzt der Agent diese Anforderung durch

und verweigert den Start, wenn die Service-Binärdatei nicht unter ist. `C:\ProgramData\Amazon\CodeDeploy\`

Themen

- Verwenden von Systems Manager
- Verwenden Sie Windows PowerShell
- Verwenden eines Direkt-Links
- Verwenden Sie einen Amazon S3 S3-Kopierbefehl

Verwenden von Systems Manager

Folgen Sie den Anweisungen unterInstallieren Sie den CodeDeploy Agenten mit AWS Systems Manager, um den Agenten zu installieren. CodeDeploy

Verwenden Sie Windows PowerShell

Melden Sie sich bei der Instanz an und führen Sie die folgenden Befehle in Windows aus PowerShell:

1. Fordern Sie an, dass nur Skripte und Konfigurationsdateien von vertrauenswürdigen Anbietern aus dem Internet heruntergeladen werden. Wenn Sie dazu aufgefordert werden, die Ausführungsrichtlinie zu ändern, geben Sie "**Y**" ein.

   ```
   Set-ExecutionPolicy RemoteSigned
   ```

2. Laden Sie das AWS Tools for Windows PowerShell.

   ```
   Import-Module AWSPowerShell
   ```

3. Erstellen Sie ein Verzeichnis, in das die CodeDeploy Agenteninstallationsdatei heruntergeladen wird.

   ```
   New-Item -Path "c:\temp" -ItemType "directory" -Force
   ```

4. Konfigurieren Sie die AWS Anmeldeinformationen mit den `Initialize-AWSDefaultConfiguration` Befehlen `Set-AWSCredential` und. Weitere Informationen finden Sie unter Verwenden von AWS Anmeldeinformationen im PowerShell Benutzerhandbuch zu AWS Tools.

5. Laden Sie die CodeDeploy Agenten-Installationsdatei herunter.

> **Note**
>
> Die neueste veröffentlichte Version finden Sie unterVersionsverlauf des Agenten CodeDeploy.

Um die neueste Version des CodeDeploy Agenten zu installieren. Das `latestv2/` Präfix dient dem Installationsprogramm der Version 2.x (2.0.x und höher), und das `latest/` Präfix dient dem Installationsprogramm der Version 1.8.x.

- Für Version 2.0.x und höher:

```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latestv2/
codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi
```

Für Version 1.8.x und früher:

```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latest/
codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi
```

Um eine bestimmte Version des Agenten zu installieren: CodeDeploy

-
```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key releases/
codedeploy-agent-###.msi -File c:\temp\codedeploy-agent.msi
```

*bucket-name*ist der Name des Amazon S3 S3-Buckets, der die CodeDeploy Resource Kit-Dateien für Ihre Region enthält. Ersetzen Sie beispielsweise für die Region USA Ost (Ohio) *bucket-name* durch`aws-codedeploy-us-east-2`. Eine Liste der Bucket-Namen finden Sie unterBucket-Namen für das Resource Kit nach Regionen.

6. Führen Sie die CodeDeploy Agenten-Installationsdatei aus.

```
c:\temp\codedeploy-agent.msi /quiet /l c:\temp\host-agent-install-log.txt
```

Führen Sie den folgenden Befehl aus, um zu prüfen, ob der Dienst ausgeführt wird:

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Wenn der CodeDeploy Agent gerade installiert und noch nicht gestartet wurde, sollten Sie nach der Ausführung des Get-Service Befehls unter Status Folgendes sehen**`Start...`**:

```
Status     Name               DisplayName
------     ----               -----------
Start...   codedeployagent    CodeDeploy Host Agent Service
```

Wenn der CodeDeploy Agent bereits läuft, sollten Sie nach der Ausführung des Get-Service Befehls unter Status Folgendes sehen**`Running`**:

```
Status     Name               DisplayName
------     ----               -----------
Running    codedeployagent    CodeDeploy Host Agent Service
```

Verwenden eines Direkt-Links

Wenn die Sicherheitseinstellungen des Browsers auf der Windows Server-Instanz die erforderlichen Berechtigungen vorsehen (z. B. für`https://s3.*.amazonaws.com`), können Sie einen direkten Link für Ihre Region verwenden, um den CodeDeploy Agenten herunterzuladen und dann das Installationsprogramm manuell auszuführen.

Das `latestv2/` Präfix dient dem Installationsprogramm der Version 2.x (2.0.x und höher), und das `latest/` Präfix dient dem Installationsprogramm der Version 1.8.x.

Für Version 2.0.x und höher lautet der Link:

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latestv2/codedeploy-agent.msi
```

Für Version 1.8.x und früher:

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latest/codedeploy-agent.msi
```

... wo *`region`* ist die AWS Region, in der Sie Ihre Anwendung bereitstellen.

Zum Beispiel für Version 2.0.x und höher:

```
https://s3.af-south-1.amazonaws.com/aws-codedeploy-af-south-1/latestv2/codedeploy-agent.msi
```

> **Important**
>
> Besorgen Sie sich die `.msi` Datei aus derselben Region wie Ihre CodeDeploy Anwendung. Wenn Sie eine andere Region wählen, kann es zu `inconsistent region` Fehlern in der `codedeploy-agent-log` Datei kommen, wenn Sie die `.msi` Datei ausführen.

Verwenden Sie einen Amazon S3 S3-Kopierbefehl

Wenn der auf der Instance installiert AWS CLI ist, können Sie den CodeDeploy Agenten mit dem Amazon S3 S3-Befehl cp herunterladen und dann das Installationsprogramm manuell ausführen. Weitere Informationen finden Sie unter Installieren von AWS Command Line Interface unter Microsoft Windows.

Das `latestv2/` Präfix dient dem Installationsprogramm der Version 2.x (2.0.x und höher), und das `latest/` Präfix dient dem Installationsprogramm der Version 1.8.x.

Für Version 2.0.x und höher lautet der Amazon S3 S3-Befehl:

```
aws s3 cp s3://aws-codedeploy-region/latestv2/codedeploy-agent.msi codedeploy-agent.msi
 --region region
```

Für Version 1.8.x und früher:

```
aws s3 cp s3://aws-codedeploy-region/latest/codedeploy-agent.msi codedeploy-agent.msi
 --region region
```

... wo *`region`* ist die AWS Region, in der Sie Ihre Anwendung bereitstellen.

Zum Beispiel für Version 2.0.x und höher:

```
aws s3 cp s3://aws-codedeploy-af-south-1/latestv2/codedeploy-agent.msi codedeploy-agent.msi --region af-south-1
```

Installieren Sie den CodeDeploy Agenten für macOS

Melden Sie sich bei der Instance an, und führen Sie die folgenden Befehle nacheinander aus.

Wechseln Sie in Ihr Home-Verzeichnis:

```
cd ~
```

Laden Sie das CodeDeploy Agent-Installationspaket herunter:

```
curl -O https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/codedeploy-agent.aarch64.pkg
```

*bucket-name*ist der Name des Amazon S3 S3-Buckets, der die CodeDeploy Resource Kit-Dateien für Ihre Region enthält, und *region-identifier* ist die Kennung für Ihre Region.

Beispiel:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/codedeploy-agent.aarch64.pkg`

Eine Liste der Bucket-Namen und Region-IDs finden Sie unterBucket-Namen für das Resource Kit nach Regionen.

Gehen Sie wie folgt vor, um die neueste Version des CodeDeploy Agenten zu installieren:

- ```
  sudo installer -pkg codedeploy-agent.aarch64.pkg -target /
  ```

Gehen Sie wie folgt vor, um eine bestimmte Version des CodeDeploy Agenten zu installieren:

- Listet die verfügbaren Versionen in Ihrer Region auf:

  ```
  aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier
   | grep '\.pkg$'
  ```

- Laden Sie eine der folgenden Versionen herunter und installieren Sie sie:

  ```
  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 /
  ```

  Die neueste veröffentlichte Version finden Sie unterVersionsverlauf des Agenten CodeDeploy.
  ```

Führen Sie den folgenden Befehl aus, um zu prüfen, ob der Dienst ausgeführt wird:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Wenn der Agent läuft, sollte eine Meldung wie `The AWS CodeDeploy agent is running.`

Wenn der Agent nicht läuft, starten Sie den Dienst und überprüfen Sie den Status:

```
sudo launchctl kickstart system/com.amazon.aws.codedeploy-agent
```

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Aktualisieren Sie den CodeDeploy Agenten

Sie können automatische, geplante Updates des CodeDeploy Agenten auf allen unterstützten Betriebssystemen mit konfigurieren AWS Systems Manager. Sie können Aktualisierungen unter allen unterstützten Betriebssystemen auch erzwingen, indem Sie einen Befehl auf einer Instance ausführen.

Themen

- Aktualisieren Sie den CodeDeploy Agenten auf Amazon Linux oder RHEL
- Aktualisieren Sie den CodeDeploy Agenten auf Ubuntu Server
- Aktualisieren Sie den CodeDeploy Agenten auf Windows Server
- Aktualisieren Sie den CodeDeploy Agenten auf macOS

Aktualisieren Sie den CodeDeploy Agenten auf Amazon Linux oder RHEL

Um automatische, geplante Updates des CodeDeploy Agenten mithilfe von zu konfigurieren AWS Systems Manager, folgen Sie den Schritten unter CodeDeploy Agent installieren mit AWS Systems Manager.

Führen Sie für Version 2.0.x und höher Folgendes aus, um ein Update zu erzwingen:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

Alternativ können Sie das regionale `latestv2/install` Skript erneut herunterladen und ausführen. Installationsanweisungen finden Sie unter Installieren Sie den CodeDeploy Agenten für Amazon Linux oder RHEL.

Führen Sie für Version 1.8.x und frühere Versionen Folgendes aus, um ein Update zu erzwingen:

```
sudo /opt/codedeploy-agent/bin/install auto
```

> **Important**
>
> Derzeit gibt es keinen automatischen Aktualisierungspfad von Version 1.8.x auf 2.0.0. Um auf 2.0.0 zu aktualisieren, installieren Sie es mit dem `AWSCodeDeployAgentV2` Systems Manager Distributor-Paket oder führen Sie das regionale `latestv2/install` Skript manuell aus. Detaillierte Anweisungen finden Sie unter Installieren Sie den CodeDeploy Agenten.

Aktualisieren Sie den CodeDeploy Agenten auf Ubuntu Server

Um automatische, geplante Updates des CodeDeploy Agenten mithilfe von zu konfigurieren AWS Systems Manager, folgen Sie den Schritten unter CodeDeploy Agent installieren mit AWS Systems Manager.

Führen Sie für Version 2.0.x und höher Folgendes aus, um ein Update zu erzwingen:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

Alternativ können Sie das regionale `latestv2/install` Skript erneut herunterladen und ausführen. Installationsanweisungen finden Sie unter Installieren Sie den CodeDeploy Agenten für Ubuntu Server.

Führen Sie für Version 1.8.x und frühere Versionen Folgendes aus, um ein Update zu erzwingen:

```
sudo /opt/codedeploy-agent/bin/install auto
```

> **Important**
>
> Derzeit gibt es keinen automatischen Aktualisierungspfad von Version 1.8.x auf 2.0.0. Um auf 2.0.0 zu aktualisieren, installieren Sie es mit dem `AWSCodeDeployAgentV2` Systems Manager Distributor-Paket oder führen Sie das regionale `latestv2/install` Skript manuell aus. Detaillierte Anweisungen finden Sie unter Installieren Sie den CodeDeploy Agenten.

Aktualisieren Sie den CodeDeploy Agenten auf Windows Server

Sie können automatische Updates des CodeDeploy Agenten mit aktivieren AWS Systems Manager. Mit Systems Manager können Sie einen Aktualisierungszeitplan für Ihre Amazon EC2- oder lokalen Instances konfigurieren, indem Sie eine Verknüpfung mit Systems Manager State Manager erstellen. Sie können den CodeDeploy Agenten auch manuell aktualisieren, indem Sie die aktuelle Version deinstallieren und eine neuere installieren.

Themen

- Richten Sie das automatische CodeDeploy Agenten-Update ein mit AWS Systems Manager
- Aktualisieren Sie den CodeDeploy Agenten manuell

Richten Sie das automatische CodeDeploy Agenten-Update ein mit AWS Systems Manager

Folgen Sie den Anweisungen unter Installieren des CodeDeploy Agenten mit, um Systems Manager zu konfigurieren und automatische Updates des CodeDeploy Agenten zu aktivieren AWS Systems Manager.

Aktualisieren Sie den CodeDeploy Agenten manuell

Um den CodeDeploy Agenten manuell zu aktualisieren, können Sie die neueste Version über die CLI oder mit Systems Manager installieren. Folgen Sie den Anweisungen unter Installieren des CodeDeploy-Agenten. Es wird empfohlen, ältere Versionen des CodeDeploy Agenten zu deinstallieren, indem Sie den Anweisungen unter CodeDeployAgent deinstallieren folgen.

Aktualisieren Sie den CodeDeploy Agenten auf macOS

Um automatische, geplante Updates des CodeDeploy Agenten mithilfe von zu konfigurieren AWS Systems Manager, folgen Sie den Schritten unter Installieren Sie den CodeDeploy Agenten mit AWS Systems Manager und wählen Sie das `AWSCodeDeployAgentV2` Paket aus.

Um ein Update zu erzwingen, führen Sie folgenden Befehl aus:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

Sie können auch die neueste `.pkg` Datei herunterladen und sie über den vorhandenen Agenten installieren. Installationsanweisungen finden Sie unter Installieren Sie den CodeDeploy Agenten für macOS.

Deinstallieren Sie den CodeDeploy Agenten

Sie können den CodeDeploy Agenten aus Instanzen entfernen, wenn er nicht mehr benötigt wird oder wenn Sie eine Neuinstallation durchführen möchten.

Deinstallieren Sie den CodeDeploy Agenten von Amazon Linux oder RHEL

Um den CodeDeploy Agenten zu deinstallieren, melden Sie sich bei der Instance an und führen Sie den folgenden Befehl aus:

```
sudo yum erase codedeploy-agent
```

Deinstallieren Sie den CodeDeploy Agenten von Ubuntu Server

Um den CodeDeploy Agenten zu deinstallieren, melden Sie sich bei der Instanz an und führen Sie den folgenden Befehl aus:

```
sudo dpkg --purge codedeploy-agent
```

Deinstallieren Sie den CodeDeploy Agenten von Windows Server

Um den CodeDeploy Agenten zu deinstallieren, melden Sie sich bei der Instanz an und führen Sie nacheinander die folgenden drei Befehle aus:

```
wmic
product where name="CodeDeploy Host Agent" call uninstall /nointeractive
exit
```

> **Note**
>
> Unter Windows Server 2025 und höher `wmic` ist es veraltet. Verwenden Sie stattdessen den folgenden PowerShell Befehl:
>
> ```
> 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 }
> ```

Sie können sich auch bei der Instanz anmelden und in der Systemsteuerung Programme und Funktionen öffnen, CodeDeploy Host Agent und anschließend Deinstallieren auswählen.

Den CodeDeploy Agenten von macOS deinstallieren

Um den CodeDeploy Agenten zu deinstallieren, melden Sie sich bei der Instanz an und führen Sie den folgenden Befehl aus:

```
sudo /opt/codedeploy-agent/bin/uninstall.sh
```

CodeDeploy Agentenprotokolle senden an CloudWatch

Sie können Metrik- und Protokolldaten für CodeDeploy Agenten an CloudWatch den Unified CloudWatch Agent oder, einfacher gesagt, an den CloudWatch Agenten senden.

Verwenden Sie die folgenden Anweisungen, um den CloudWatch Agenten zu installieren und ihn für die Verwendung mit CodeDeploy Agenten zu konfigurieren.

Voraussetzungen

Führen Sie als Erstes die folgenden Schritte aus:

- Installieren Sie den CodeDeploy Agenten und stellen Sie sicher, dass er läuft. Weitere Informationen erhalten Sie unter Installieren Sie den CodeDeploy Agenten und Stellen Sie sicher, dass der CodeDeploy Agent läuft.
- Installieren Sie den CloudWatch Agenten. Weitere Informationen finden Sie unter Installation des CloudWatch Agenten.
- Fügen Sie dem CodeDeploy IAM-Instanzprofil die folgenden Berechtigungen hinzu:
  - CloudWatchLogsFullAccess
  - CloudWatchAgentServerPolicy

  Weitere Informationen zum CodeDeploy Instanzprofil finden Sie unter Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances vonErste Schritte mit CodeDeploy.

Konfigurieren Sie den CloudWatch Agenten für die Erfassung von CodeDeploy Protokollen

Sie können den CloudWatch Agenten konfigurieren, indem Sie einen Assistenten ausführen oder eine Konfigurationsdatei manuell erstellen oder bearbeiten.

So konfigurieren Sie den CloudWatch Agenten mit dem Assistenten (Linux)

1. Führen Sie den Assistenten aus, wie unter Den Assistenten zur CloudWatch Agentenkonfiguration ausführen beschrieben.
2. Geben Sie im Assistenten die `Do you want to monitor any log files?` Eingabetaste ein, wenn Sie dazu aufgefordert **1** werden.
3. Geben Sie die CodeDeploy Agent-Protokolldatei wie folgt an:
   1. `Log file path`Geben Sie den Pfad für die CodeDeploy Protokolldatei ein, zum Beispiel:**/var/log/aws/codedeploy-agent/codedeploy-agent.log**.
   2. `Log group name`Geben Sie einen Namen für die Protokollgruppe ein, zum Beispiel:**codedeploy-agent-log**.
   3. `Log stream name`Geben Sie einen Namen für den Protokollstream ein, zum Beispiel:**{instance_id}-codedeploy-agent-log**.
4. Wenn Sie dazu aufgefordert werden`Do you want to specify any additional log files?`, geben Sie ein**1**.
5. Geben Sie die CodeDeploy Agenten-Bereitstellungsprotokolle wie folgt an:
   1. `Log file path`Geben Sie zum Beispiel den Pfad für die CodeDeploy Bereitstellungsprotokolldatei ein:**/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log**.
   2. `Log group name`Geben Sie einen Namen für die Protokollgruppe ein, zum Beispiel:**codedeploy-agent-deployment-log**.
   3. `Log stream name`Geben Sie einen Namen für den Protokollstream ein, zum Beispiel:**{instance_id}-codedeploy-agent-deployment-log**.
6. Wenn Sie dazu aufgefordert werden`Do you want to specify any additional log files?`, geben Sie ein**1**.
7. Geben Sie die CodeDeploy Agent-Updater-Protokolle wie folgt an:
   1. Geben Sie für `Log file path` den Pfad für die CodeDeploy Updater-Protokolldatei ein. Der Pfad hängt von Ihrer Agentenversion ab:
      - Verwenden **/var/log/aws/codedeploy-agent/codedeploy-agent-updater.log** Sie für Version 2.0.x und höher.
      - Verwenden Sie für Version 1.8.x und früher. **/tmp/codedeploy-agent.update.log**

2. `Log group name`Geben Sie einen Protokollgruppennamen ein, zum Beispiel:. **`codedeploy-agent-updater-log`**
3. `Log stream name`Geben Sie einen Namen für den Protokollstream ein, zum Beispiel:**`{instance_id}-codedeploy-agent-updater-log`**.

Um den CloudWatch Agenten mit dem Assistenten zu konfigurieren (Windows)

1. Führen Sie den Assistenten aus, wie unter Den Assistenten zur CloudWatch Agentenkonfiguration ausführen beschrieben.
2. Geben Sie im Assistenten die `Do you want to monitor any customized log files?` Eingabetaste ein, wenn Sie dazu aufgefordert **1** werden.
3. Geben Sie die CodeDeploy Protokolldatei wie folgt an:
   1. Geben Sie für `Log file path` den Pfad für die CodeDeploy Agenten-Protokolldatei ein. Der Pfad hängt von Ihrer Agentenversion ab:
      - Verwenden **`C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log`** Sie für Version 2.0.x und höher.
      - Verwenden Sie für Version 1.8.x und früher. **`C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt`**
   2. `Log group name`Geben Sie einen Protokollgruppennamen ein, zum Beispiel:. **`codedeploy-agent-log`**
   3. `Log stream name`Geben Sie einen Namen für den Protokollstream ein, zum Beispiel:**`{instance_id}-codedeploy-agent-log`**.
4. Wenn Sie dazu aufgefordert werden`Do you want to specify any additional log files?`, geben Sie ein**1**.
5. Geben Sie die CodeDeploy Agenten-Bereitstellungsprotokolle wie folgt an:
   1. Geben Sie für `Log file path` den Pfad für die CodeDeploy Bereitstellungsprotokolldatei ein:**`C:\ProgramData\Amazon\CodeDeploy\deployment-logs\codedeploy-agent-deployments.log`**.
   2. `Log group name`Geben Sie zum Beispiel einen Namen für die Protokollgruppe ein:**`codedeploy-agent-deployment-log`**.
   3. `Log stream name`Geben Sie einen Namen für den Protokollstream ein, zum Beispiel:**`{instance_id}-codedeploy-agent-deployment-log`**.

Um den CloudWatch Agenten durch manuelles Erstellen oder Bearbeiten einer Konfigurationsdatei zu konfigurieren (Linux)

1. Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei wie unter Manuelles Erstellen oder Bearbeiten der CloudWatch Agenten-Konfigurationsdatei beschrieben.
2. Stellen Sie sicher, dass die Datei aufgerufen wird `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` und den folgenden Code enthält:

   Der Protokollpfad des Updaters unterscheidet sich je nach Version. Verwenden Sie für Version 2.0.x und höher. `/var/log/aws/codedeploy-agent/codedeploy-agent-updater.log` Verwenden Sie für Version 1.8.x und früher. `/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"
                }
            ]
        }
    }
}
...
```

Um den CloudWatch Agenten durch manuelles Erstellen oder Bearbeiten einer Konfigurationsdatei zu konfigurieren (Windows)

1. Erstellen oder bearbeiten Sie die CloudWatch Agent-Konfigurationsdatei wie unter Manuelles Erstellen oder Bearbeiten der CloudWatch Agenten-Konfigurationsdatei beschrieben.
2. Stellen Sie sicher, dass die Datei aufgerufen wird `C:\ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json` und den folgenden Code enthält:

   Für Version 2.0.x und höher lautet `codedeploy-agent.log` der Dateiname des Agent-Protokolls. Für Version 1.8.x und früher lautet der Dateiname des Agent-Protokolls. `codedeploy-agent-log.txt` Der Pfad des Bereitstellungsprotokolls ist für beide Versionen derselbe.

```
...
"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"
                    }
                ]
            },
            ...
        }
    },
...
```

Starten Sie den CloudWatch Agenten neu

Nachdem Sie Ihre Änderungen vorgenommen haben, starten Sie den CloudWatch Agenten neu, wie unter CloudWatch Agent starten beschrieben.

Arbeiten mit Instanzen für CodeDeploy

CodeDeploy unterstützt Bereitstellungen auf Instances, auf denen Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) und Windows Server ausgeführt werden.

Sie können CodeDeploy die Bereitstellung sowohl für Amazon EC2 EC2-Instances als auch für lokale Instances verwenden. Eine lokale Instance ist jedes physische Gerät, bei dem es sich nicht um eine Amazon EC2 EC2-Instance handelt, auf dem der CodeDeploy Agent ausgeführt und eine Verbindung zu öffentlichen AWS Service-Endpunkten hergestellt werden kann. Sie können CodeDeploy damit eine Anwendung gleichzeitig auf Amazon EC2 EC2-Instances in der Cloud und auf Desktop-PCs in Ihrem Büro oder Servern in Ihrem eigenen Rechenzentrum bereitstellen.

Vergleich von Amazon EC2 EC2-Instances mit lokalen Instances

In der folgenden Tabelle werden Amazon EC2 EC2-Instances und lokale Instances verglichen:

Betreff	Amazon EC2-Instances	On-premises -Instances
Erfordert, dass Sie eine Version des CodeDeploy Agenten installieren und ausführen, die mit dem Betriebssystem kompatibe l ist, das auf der Instance ausgeführt wird.	Ja	Ja
Erfordern, dass die jeweilige Instance eine Verbindung zum CodeDeploy-Service herstelle n kann.	Ja	Ja
Erfordert, dass ein IAM-Instanzprofil an die Instanz angehängt wird. Das IAM-Instanzprofil muss über Berechtigungen verfügen, um an Bereitstellungen teilnehme	Ja	Nein

Betreff	Amazon EC2-Instances	On-premises -Instances
n CodeDeploy zu können. Weitere Informationen finden Sie unter Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances.		
Erfordern einen der folgenden Schritte zur Authentifizierung und Registrierung von Instances: • Erstellen Sie eine IAM-Rolle, die von einem IAM-Benutzer auf jeder Instance übernommen werden kann, um regelmäßig aktualisierte temporäre Anmeldeinformationen abzurufen, die über generiert wurden. AWS-Security-Token-Service • Erstellen Sie einen IAM-Benutzer für jede Instanz und speichern Sie die Kontoanmeldeinformationen des IAM-Benutzers in Klartext auf der Instanz.	Nein	Ja
Erfordert, dass Sie jede Instanz bei registrieren, CodeDeploy bevor Sie sie bereitstellen können.	Nein	Ja

Betreff	Amazon EC2-Instances	On-premises -Instances
Erfordert, dass Sie jede Instanz taggen, bevor Sie sie bereitstellen CodeDeploy können.	Ja	Ja
Kann im Rahmen von CodeDeploy Bereitstellungen an Amazon EC2 Auto Scaling- und Elastic Load Balancing Balancing-Szenarien teilnehmen.	Ja	Nein
Kann aus Amazon S3 S3-Buckets und GitHub Repositorys bereitgestellt werden.	Ja	Ja
Unterstützen Auslöser für das Senden von SMS oder E-Mail-Benachrichtigungen bei Eintreten bestimmter Ereignisse in Bereitstellungen oder Instances.	Ja	Ja
Werden für zugeordnete Bereitstellungen mit Kosten belastet.	Nein	Ja

Instanzaufgaben für CodeDeploy

Befolgen Sie die unten angegebenen Anweisungen, um Instances für Bereitstellungen zu starten oder zu konfigurieren:

Ich möchte eine neue Amazon Linux- oder Windows Server Amazon EC2 EC2-Instance starten.	Informationen zum Starten der Amazon EC2 EC2-Instance mit dem geringsten Aufwand finden Sie unterErstellen Sie eine Amazon EC2

	EC2-Instance für CodeDeploy (CloudFormation Vorlage). Informationen zum Starten der Amazon EC2 EC2-Instance größtenteils selbst finden Sie unterErstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy (AWS CLI oder Amazon EC2 EC2-Konsole).
Ich möchte eine neue Ubuntu Server- oder RHEL Amazon EC2 EC2-Instance starten.	Siehe Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy (AWS CLI oder Amazon EC2 EC2-Konsole).
Ich möchte eine Amazon Linux-, Windows Server-, Ubuntu Server- oder RHEL Amazon EC2 EC2-Instance konfigurieren.	Siehe Konfigurieren Sie eine Amazon EC2 EC2-Instance, mit der Sie arbeiten können CodeDeploy.
Ich möchte eine lokale Windows Server-, Ubuntu Server- oder RHEL-Instance konfigurieren (physische Geräte, die keine Amazon EC2 EC2-Instances sind).	Siehe Working with On-Premises Instances.
Ich CodeDeploy möchte während einer Bereitstellung eine Ersatzflotte von Instances bereitstellen. blue/green	Siehe Arbeiten mit Bereitstellungen in CodeDeploy.

Um Amazon EC2-Instances in Amazon EC2 Auto Scaling-Gruppen vorzubereiten, müssen Sie einige zusätzliche Schritte ausführen. Weitere Informationen finden Sie unter Integration CodeDeploy mit Amazon EC2 Auto Scaling.

Topics

- Tagging Instances for Deployments
- Working with Amazon EC2 Instances
- Working with On-Premises Instances
- View Instance Details
- Instance Health

Tagging von Instanzen für Bereitstellungsgruppen in CodeDeploy

Um Ihre Amazon EC2 EC2-Instances und lokalen Instances zu verwalten, können Sie Tags verwenden, um jeder Ressource Ihre eigenen Metadaten zuzuweisen. Tags ermöglichen Ihnen, Ihre Instances auf unterschiedliche Weise zu kategorisieren (z. B. nach Zweck, Eigentümer oder Umgebung). Dies ist nützlich, wenn Sie viele Instances haben. Sie können eine Instance oder eine Gruppe von Instances schnell auf der Grundlage der Tags identifizieren, die Sie ihnen zugewiesen haben. Jeder Tag (Markierung) besteht aus einem Schlüssel und einem optionalen Wert, beides können Sie bestimmen. Weitere Informationen finden Sie unter Taggen Ihrer Amazon EC2 EC2-Ressourcen.

Um anzugeben, welche Instances in einer CodeDeploy Bereitstellungsgruppe enthalten sind, geben Sie Tags in einer oder mehreren Tag-Gruppen an. Instances, die Ihren Tag-Kriterien entsprechen, sind diejenigen, auf denen die letzte Anwendungsversion installiert wird, wenn eine Bereitstellung zu dieser Bereitstellungsgruppe erstellt wird.

> **Note**
>
> Sie können Amazon EC2 Auto Scaling Scaling-Gruppen auch in Bereitstellungsgruppen aufnehmen, sie werden jedoch anhand ihrer Namen und nicht anhand von Tags identifiziert, die auf Instances angewendet werden. Weitere Informationen finden Sie unter Integration CodeDeploy mit Amazon EC2 Auto Scaling.

Die Kriterien für die Instances in einer Bereitstellungsgruppe können so einfach wie ein einzelnes Tag in einer einzigen Tag-Gruppe sein. Sie können auch so komplex wie jeweils 10 Tags in maximal drei Tag-Gruppen sein.

Wenn Sie eine einzelne Tag-Gruppe verwenden, ist jede von mindestens einem Tag in der Gruppe identifizierte Instance in der Bereitstellungsgruppe enthalten. Wenn Sie mehrere Tag-Gruppen verwenden, sind nur die Instances, die durch mindestens einen Tag in jeder Tag-Gruppe identifiziert werden, enthalten.

Die folgenden Beispiele veranschaulichen, wie Tags und Tag-Gruppen verwendet werden können, um die Instances für eine Bereitstellungsgruppe auszuwählen.

Themen

- Beispiel 1: Einzelne Tag-Gruppe, einzelnes Tag

- Beispiel 2: Einzelne Tag-Gruppe, mehrere Tags
- Beispiel 3: Mehrere Tag-Gruppen, einzelne Tags
- Beispiel 4: Mehrere Tag-Gruppen, mehrere Tags

Beispiel 1: Einzelne Tag-Gruppe, einzelnes Tag

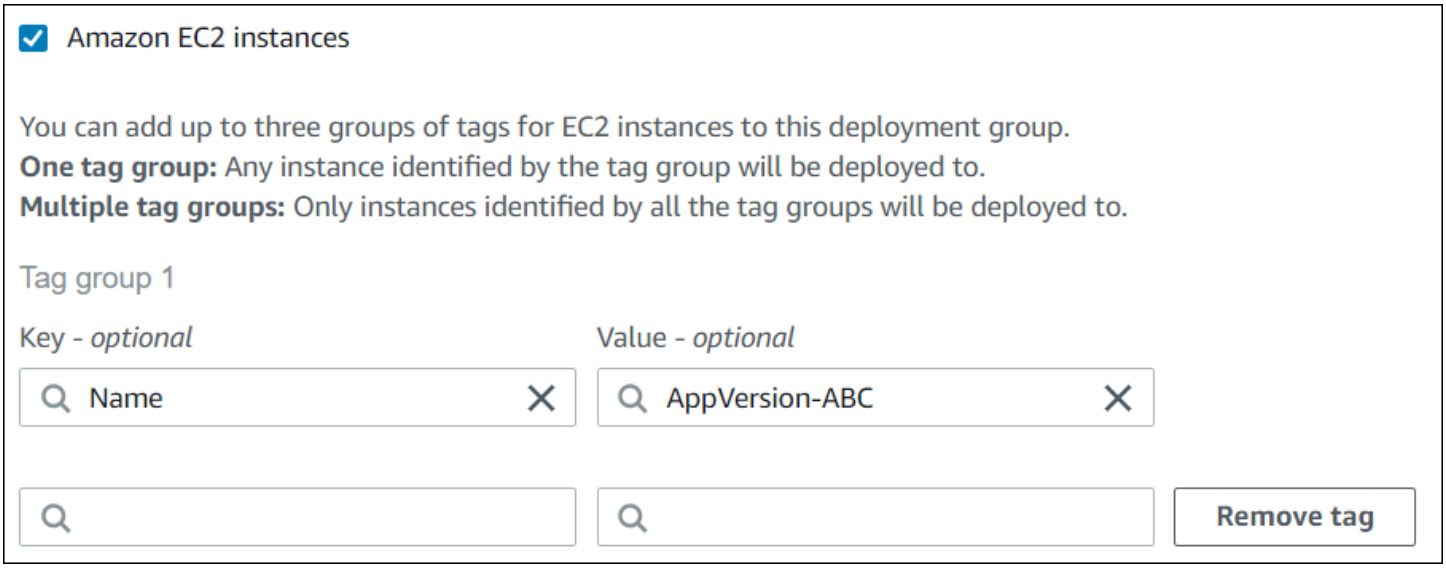
Sie können einen einzelnen tag in einer einzelnen Tag-Gruppe angeben:

Tag-Gruppe 1

Key (Schlüssel)	Value (Wert)
Name	AppVersion-ABC

Jede Instance, die mit `Name=AppVersion-ABC` markiert ist, ist Teil der Bereitstellungsgruppe, selbst wenn sie noch weitere Tags hat.

CodeDeploy Ansicht des Konsolen-Setups:



JSON-Struktur:

```
"ec2TagSet": {
        "ec2TagSetList": [
           [
              {
```

```
                "Type": "KEY_AND_VALUE",
                "Key": "Name",
                "Value": "AppVersion-ABC"
            }
        ]
    ]
},
```

Beispiel 2: Einzelne Tag-Gruppe, mehrere Tags

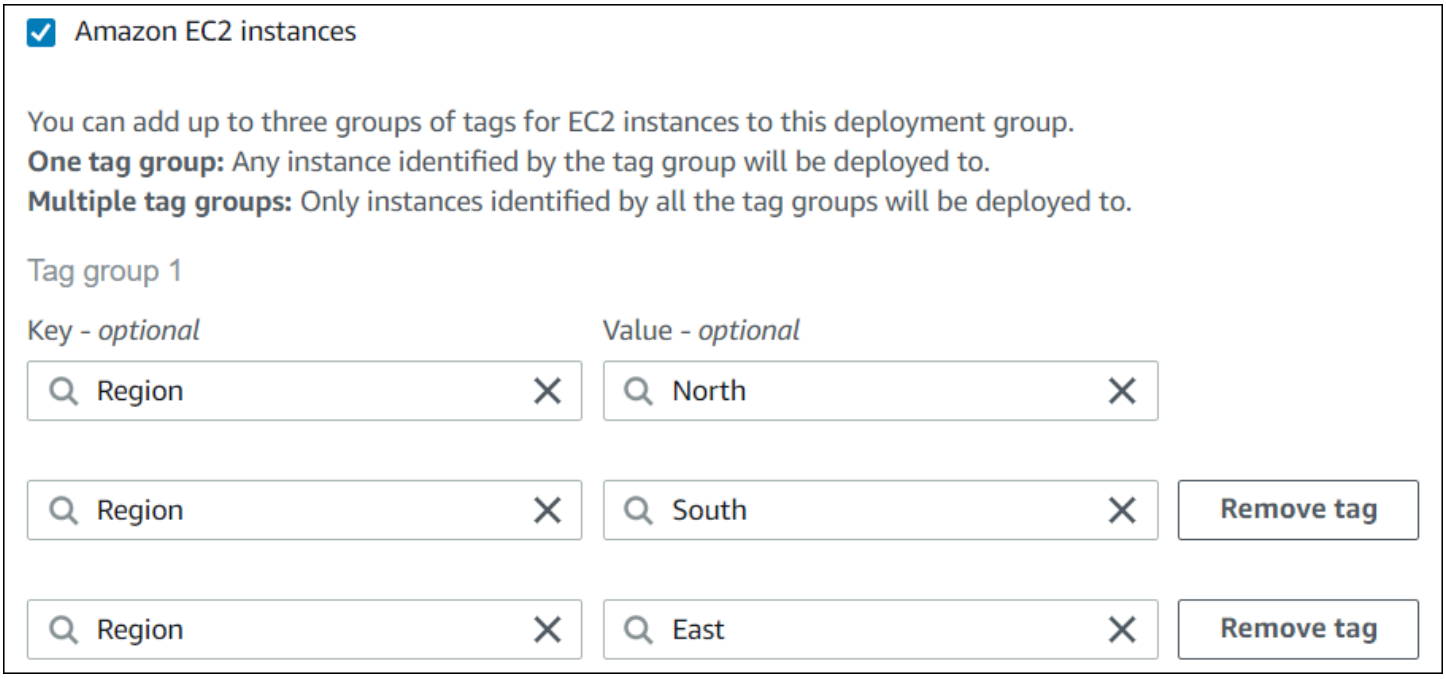
Sie können auch mehrere Tags in einer einzigen Tag-Gruppe angeben:

Tag-Gruppe 1

Key (Schlüssel)	Value (Wert)
Region	Nord
Region	Süd
Region	Ost

Eine Instance, die mit einem dieser drei Tags markiert ist, ist Teil der Bereitstellungsgruppe, selbst wenn sie noch weitere Tags hat. Wenn Sie beispielsweise weitere mit `Region=West` markierte Instances haben, sind diese nicht Teil der Bereitstellungsgruppe.

CodeDeploy Ansicht des Konsolen-Setups:



JSON-Struktur:

```
"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"
               }
            ]
         ]
      },
```

Beispiel 3: Mehrere Tag-Gruppen, einzelne Tags

Sie können auch mehrere Sätze von Tag-Gruppen mit jeweils einem einzelnen Schlüssel-Wert-Paar verwenden, um die Kriterien für Instances in der Bereitstellungsgruppe anzugeben. Wenn Sie mehrere Tag-Gruppen in einer Bereitstellungsgruppe verwenden, gehören nur Instances, die von allen tag-Gruppen identifiziert werden, zu der Bereitstellungsgruppe.

Tag-Gruppe 1

Key (Schlüssel)	Value (Wert)
Name	AppVersion-ABC

Tag-Gruppe 2

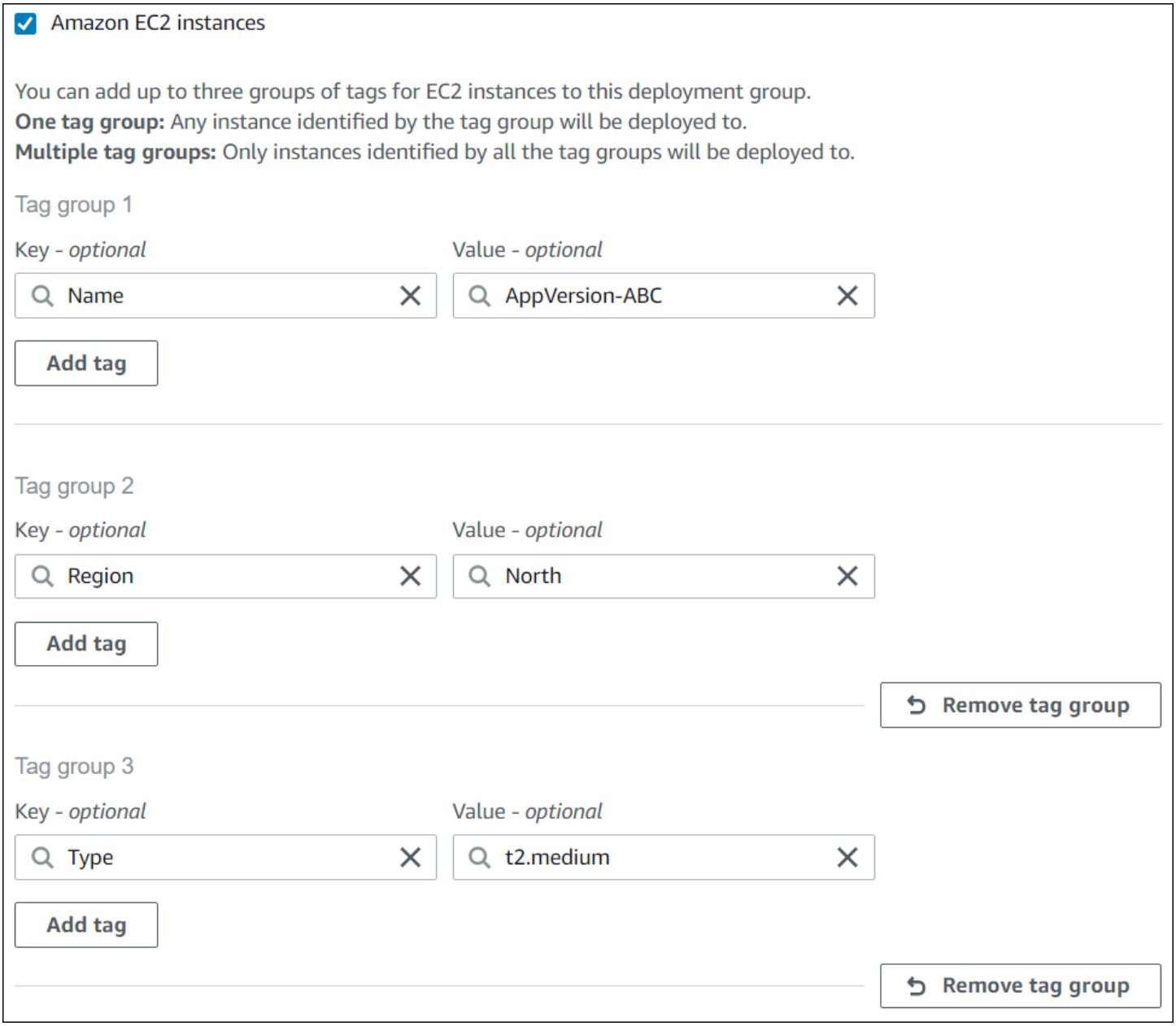
Key (Schlüssel)	Value (Wert)
Region	Nord

Gruppe 3 taggen

Key (Schlüssel)	Value (Wert)
Typ	t2.medium

Möglicherweise verfügen Sie über Instances in vielen Regionen und mit verschiedenen Instance-Typen, die mit markiert sind`Name=AppVersion-ABC`. In diesem Beispiel sind nur die Instances, die auch mit `Region=North` und `Type=t2.medium` markiert sind, Teil der Bereitstellungsgruppe.

CodeDeploy Ansicht des Konsolen-Setups:



JSON-Struktur:

```
"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"
               }
            ],
         ]
      },
```

Beispiel 4: Mehrere Tag-Gruppen, mehrere Tags

Wenn Sie mehrere Tag-Gruppen mit mehreren Tags in mindestens einer Gruppe verwenden, muss eine Instance mindestens mit einem der Tags in jeder der Gruppen übereinstimmen.

Tag-Gruppe 1

Key (Schlüssel)	Value (Wert)
Umgebung	Beta
Umgebung	Staging

Tag-Gruppe 2

Key (Schlüssel)	Value (Wert)
Region	Nord
Region	Süd
Region	Ost

Gruppe 3 taggen

Key (Schlüssel)	Value (Wert)
Typ	t2.medium
Typ	t2.large

In diesem Beispiel muss eine Instance, um Mitglied der Bereitstellungsgruppe zu sein, mit `Environment=Beta` oder `Environment=Staging`, mit (2) `Region=North`, `Region=South`oder `Region=East`, und mit (3) `Type=t2.medium` oder `Type=t2.large` markiert sein.

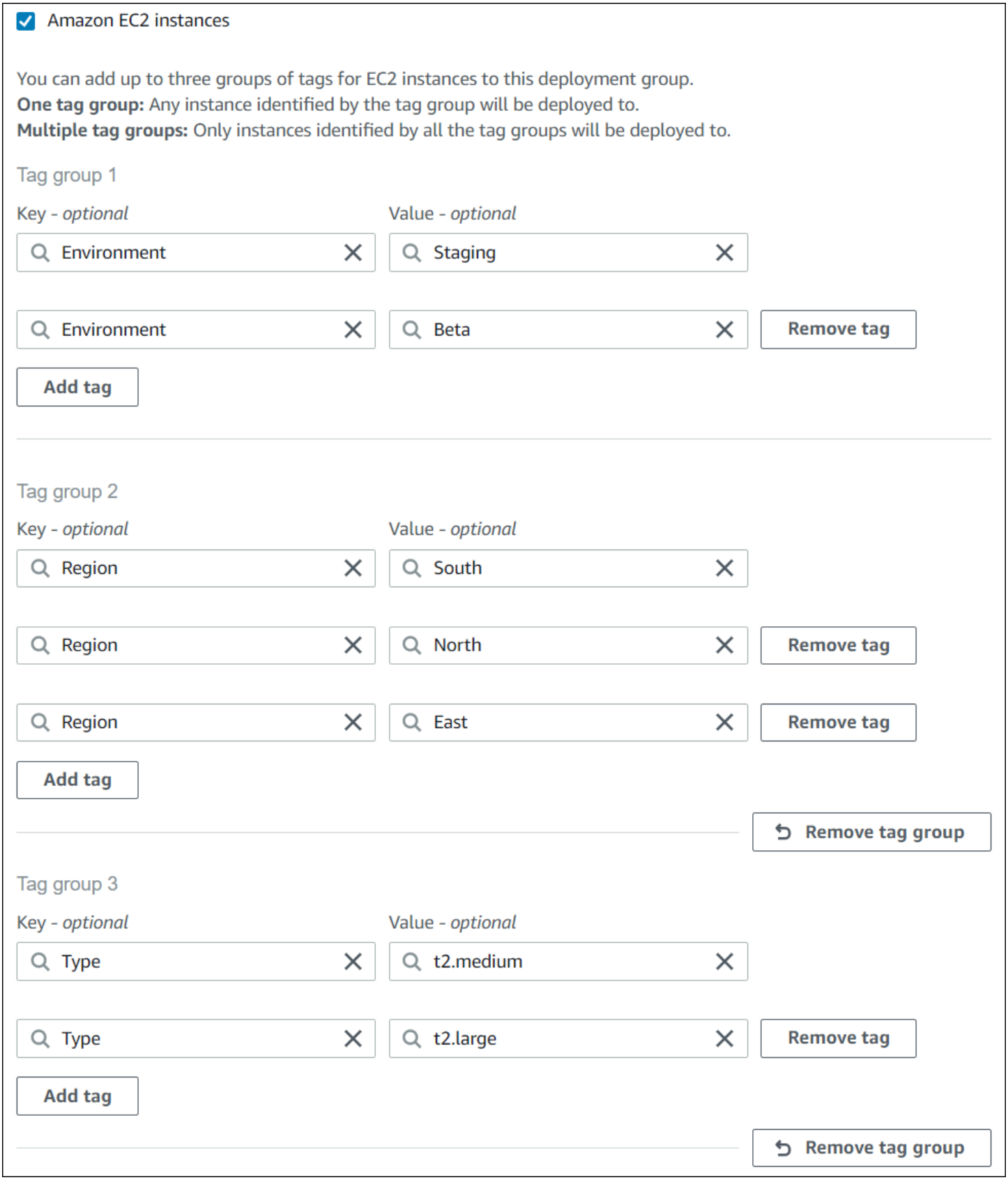
Beispielsweise sind Instances mit den folgenden Tag-Gruppen Teil der Bereitstellungsgruppe:

- `Environment=Beta`, `Region=North`,`Type=t2.medium`
- `Environment=Staging`,`Region=East`,`Type=t2.large`
- `Environment=Staging`,`Region=South`,`Type=t2.large`

Instances mit den folgenden Tag-Gruppen sind nicht Teil der Bereitstellungsgruppe. Die hervorgehobenen Schlüsselwerte führen dazu, dass die Instances ausgeschlossen werden:

- `Environment=Beta`, Region=West,`Type=t2.medium`
- `Environment=Staging`,`Region=East`,Typ=t2.micro
- Umgebung=Produktion,`Region=South`,`Type=t2.large`

CodeDeploy Ansicht des Konsolen-Setups:


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

JSON-Struktur:

```
"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"
               }
```

```
            ],
        ]
    },
```

Arbeiten mit Amazon EC2 EC2-Instances für CodeDeploy

Eine Amazon EC2 EC2-Instance ist eine virtuelle Computerumgebung, die Sie mit Amazon Elastic Compute Cloud erstellen und konfigurieren. Amazon EC2 bietet skalierbare Rechenkapazität in der AWS Cloud. Sie können Amazon EC2 verwenden, um so viele oder so wenige virtuelle Server zu starten, wie Sie für Ihre CodeDeploy Bereitstellungen benötigen.

Weitere Informationen zu Amazon EC2 finden Sie im Amazon EC2 Getting Started Guide.

Die Anweisungen in diesem Abschnitt zeigen Ihnen, wie Sie Amazon EC2 EC2-Instances für die Verwendung in Ihren CodeDeploy Bereitstellungen erstellen und konfigurieren.

Themen

- Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy (AWS CLI oder Amazon EC2 EC2-Konsole)
- Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy (CloudFormation Vorlage)
- Konfigurieren Sie eine Amazon EC2 EC2-Instance, mit der Sie arbeiten können CodeDeploy

Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy (AWS CLI oder Amazon EC2 EC2-Konsole)

Diese Anweisungen zeigen Ihnen, wie Sie eine neue Amazon EC2 EC2-Instance starten, die für die Verwendung in CodeDeploy Bereitstellungen konfiguriert ist.

Sie können unsere CloudFormation Vorlage verwenden, um eine Amazon EC2 EC2-Instance zu starten, auf der Amazon Linux oder Windows Server ausgeführt wird und die bereits für die Verwendung in CodeDeploy Bereitstellungen konfiguriert ist. Wir bieten keine CloudFormation Vorlage für Amazon EC2 EC2-Instances, auf denen Ubuntu Server oder Red Hat Enterprise Linux (RHEL) ausgeführt werden. Alternativen zur Nutzung der Vorlage finden Sie unter Arbeiten mit Instanzen für CodeDeploy.

Sie können die Amazon EC2 EC2-Konsole oder Amazon EC2 EC2-APIs verwenden AWS CLI, um eine Amazon EC2 EC2-Instance zu starten.

Starten Sie eine Amazon EC2 EC2-Instance (Konsole)

Voraussetzungen

Falls Sie dies noch nicht getan haben, folgen Sie den Anweisungen unter, Erste Schritte mit CodeDeploy um das AWS CLI einzurichten und zu konfigurieren und ein IAM-Instance-Profil zu erstellen.

Starten Sie eine Amazon-EC2-Instance

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances und Launch Instance (Instance starten) aus.
3. Suchen Sie auf der Seite Step 1: Choose an Amazon Machine Image (AMI) (Schritt 1: Auswählen eines Amazon Machine Images (AMI)) auf der Registerkarte Quick Start (Schnellstart) das Betriebssystem und die Version, die Sie verwenden möchten, und wählen Sie dann Select (Auswählen) aus. Sie müssen ein Amazon EC2 AMI-Betriebssystem wählen, das von CodeDeploy unterstützt wird. Weitere Informationen finden Sie unter Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy.
4. Wählen Sie auf der Seite Schritt 2: Instance-Typ auswählen einen beliebigen verfügbaren Amazon EC2 EC2-Instance-Typ aus und klicken Sie dann auf Weiter: Instance-Details konfigurieren.
5. Wählen Sie auf der Seite Schritt 3: Instance-Details konfigurieren in der IAM-Rollenliste die IAM-Instance-Rolle aus, in der Sie sie erstellt haben. Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances Wenn Sie den empfohlenen Rollennamen verwenden, wählen Sie **`CodeDeployDemo-EC2-Instance-Profile`** aus. Wenn Sie einen eigenen Rollenname erstellt haben, wählen Sie diesen aus.

   > **Note**
   >
   > Wenn eine standardmäßige Virtual Private Cloud (VPC) nicht in der Netzwerkliste angezeigt wird, müssen Sie eine Amazon-VPC und ein Amazon-Subnetz auswählen oder erstellen. Wählen Sie Create new VPC (Neue VPC erstellen) oder Create new subnet (Neues Subnetz erstellen) oder beides aus. Weitere Informationen finden Sie unter Ihre VPC und Subnetze.

6. Wählen Sie Next: Add Storage aus.

7. Lassen Sie die Seite Step 4: Add Storage (Schritt 4: Speicher hinzufügen) unverändert, und wählen Sie Next: Add Tags (Weiter: Tags hinzufügen) aus.
8. Wählen Sie auf der Seite Step 5: Add Tags (Schritt 5: Tags hinzufügen) die Option Add Tag (Tag hinzufügen) aus.
9. Geben Sie im Feld Schlüssel **`Name`** ein. Geben Sie im Feld Value (Wert) den Wert **`CodeDeployDemo`** ein.

   > **Important**
   >
   > Bei den Inhalten der Felder Key (Schlüssel) und Value (Wert) ist die Groß- und Kleinschreibung zu beachten.

10. Wählen Sie Weiter: Sicherheitsgruppe konfigurieren aus.
11. Lassen Sie auf der Seite Step 6: Configure Security Group (Schritt 6: Konfigurieren einer Sicherheitsgruppe) die Option Create a new security group (Neue Sicherheitsgruppe erstellen) ausgewählt.

    Eine Standard-SSH-Rolle ist für Amazon EC2 EC2-Instances konfiguriert, auf denen Amazon Linux, Ubuntu Server oder RHEL ausgeführt wird. Eine standardmäßige RDP-Rolle ist für Amazon EC2 EC2-Instances konfiguriert, auf denen Windows Server ausgeführt wird.
12. Wenn Sie den HTTP-Port öffnen möchten, klicken Sie auf die Schaltfläche Add Rule (Regel hinzufügen). Wählen Sie in der Dropdown-Liste Type (Typ) die Option **`HTTP`** aus. Akzeptieren Sie den Standardquellwert Custom 0.0.0. 0/0, und wählen Sie dann Überprüfen und Starten.

    > **Note**
    >
    > In einer Produktionsumgebung empfehlen wir, den Zugriff auf die SSH-, RDP- und HTTP-Ports einzuschränken, anstatt Anywhere 0.0.0 anzugeben. 0/0. CodeDeploy erfordert keinen uneingeschränkten Portzugriff und erfordert keinen HTTP-Zugriff. Weitere Informationen finden Sie unter Tipps zur Sicherung Ihrer Amazon EC2 EC2-Instance.

    Wenn das Dialogfeld Boot from General Purpose (SSD) (Neustart von Allzweck-(SSD)-Volume) angezeigt wird, befolgen Sie die Anweisungen, und wählen Sie dann Next (Weiter) aus.
13. Lassen Sie die Seite Step 7: Review Instance Launch (Schritt 7: Prüfen eines Starts einer Instance) unverändert und wählen Sie Launch (Starten) aus.

14. Wählen Sie im Dialogfeld Select an existing key pair or create a new key pair (Ein bestehendes Schlüsselpaar wählen oder ein neues Schlüsselpaar erstellen) entweder Choose an existing key pair (Ein vorhandenes Schlüsselpaars auswählen) oder Create a new key pair (Ein neues Schlüsselpaar erstellen). Wenn Sie ein Amazon EC2-Instance-Schlüsselpaar bereits konfiguriert haben, können Sie das Schlüsselpaar hier auswählen.

    Wenn Sie noch kein Amazon EC2-Instance-Schlüsselpaar haben, klicken Sie auf Create a new key pair (Ein neues Schlüsselpaar erstellen) und geben Sie einen wiedererkennbaren Namen ein. Wählen Sie key pair herunterladen, um das Amazon EC2 EC2-Instance-Schlüsselpaar auf Ihren Computer herunterzuladen.

    > **Important**
    >
    > Sie benötigen ein key pair, wenn Sie mit SSH oder RDP auf Ihre Amazon EC2 EC2-Instance zugreifen möchten.

15. Wählen Sie Instances starten aus.
16. Wählen Sie die ID für Ihre Amazon EC2 EC2-Instance. Fahren Sie nicht fort, bis die Instance gestartet wurde und alle Prüfungen bestanden hat.

Installieren Sie den Agenten CodeDeploy

Der CodeDeploy Agent muss auf Ihrer Amazon EC2 EC2-Instance installiert sein, bevor Sie ihn in CodeDeploy Bereitstellungen verwenden können. Weitere Informationen finden Sie unter Installieren Sie den CodeDeploy Agenten.

> **Note**
>
> Sie können die automatische Installation und Updates des CodeDeploy Agenten konfigurieren, wenn Sie Ihre Bereitstellungsgruppe in der Konsole erstellen.

Starten Sie eine Amazon EC2 EC2-Instance (CLI)

Voraussetzungen

Falls Sie dies noch nicht getan haben, folgen Sie den Anweisungen unter, Erste Schritte mit CodeDeploy um das AWS CLI einzurichten und zu konfigurieren und ein IAM-Instance-Profil zu erstellen.

Starten Sie eine Amazon-EC2-Instance

1. Nur für Windows Server Wenn Sie eine Amazon EC2 EC2-Instance erstellen, auf der Windows Server ausgeführt wird, rufen Sie die authorize-security-group-ingress Befehle create-security-group und auf, um eine Sicherheitsgruppe zu erstellen, die RDP-Zugriff (der standardmäßig nicht zulässig ist) und alternativ HTTP-Zugriff ermöglicht. Um beispielsweise eine Sicherheitsgruppe mit dem Namen CodeDeployDemo-Windows-Security-Group zu erstellen, rufen Sie die folgenden Befehle einzeln auf:

   ```
   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**
   >
   > Mit diesen Befehlen kann eine Sicherheitsgruppe zu Demonstrationszwecken erstellt werden, die uneingeschränkten RDP-Zugriff über Port 3389 und alternativ HTTP-Zugriff über Port 80 ermöglicht. Als bewährte Methode empfehlen wir, den Zugriff auf RDP- und HTTP-Ports einzuschränken. CodeDeploy benötigt keinen uneingeschränkten Port-Zugriff und erfordert keinen HTTP-Zugriff. Weitere Informationen finden Sie unter Tipps zur Sicherung Ihrer Amazon EC2 EC2-Instance.

2. Rufen Sie den run-instances Befehl auf, um die Amazon EC2 EC2-Instance zu erstellen und zu starten.

   Bevor Sie diesen Befehl aufrufen, müssen Sie Folgendes erfassen:

   - Die ID eines Amazon Machine Image (AMI) (*`ami-id`*), das Sie für die Instance verwenden. Informationen zum Abrufen der ID finden Sie unter Ein geeignetes AMI finden.
   - Der Name des Typs der Amazon EC2 EC2-Instance (*`instance-type`*), die Sie erstellen, z. B. `t1.micro` Eine Liste finden Sie unter Amazon EC2 EC2-Instance-Typen.

- Der Name eines IAM-Instance-Profils mit der Berechtigung für den Zugriff auf den Amazon S3 S3-Bucket, in dem die CodeDeploy Agenteninstallationsdateien für Ihre Region gespeichert sind.

  Informationen zur Erstellung eines IAM-Instance-Profils finden Sie unter. Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances

- Der Name eines Amazon EC2 EC2-Instance-Schlüsselpaars (*key-name*), um den SSH-Zugriff auf eine Amazon EC2 EC2-Instance zu ermöglichen, auf der Amazon Linux, Ubuntu Server oder RHEL- oder RDP-Zugriff auf eine Amazon EC2 EC2-Instance ausgeführt wird, auf der Windows Server ausgeführt wird.

  > **Important**
  >
  > Geben Sie nur den Namen des Schlüsselpaars ein, nicht die Dateierweiterung des Schlüsselpaars. Zum Beispiel my-keypair, nicht my-keypair.pem.

  Um einen Schlüsselpaarnamen zu finden, öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2. Wählen Sie im Navigationsbereich unter Network & Security (Netzwerk und Sicherheit) Key Pairs (Schlüsselpaare) aus und notieren Sie den Schlüsselpaarnamen in der Liste.

  Informationen zum Generieren eines key pair finden Sie unter Erstellen Ihres key pair mit Amazon EC2. Stellen Sie sicher, dass Sie das key pair in einer der unter Region aufgeführten Regionen und Endpunkten in Allgemeine AWS-Referenzerstellen. Andernfalls können Sie das Amazon EC2 EC2-Instance-Schlüsselpaar nicht mit CodeDeploy verwenden.

Für Amazon Linux, RHEL und Ubuntu Server

Um den run-instances Befehl zum Starten einer Amazon EC2 EC2-Instance aufzurufen, auf der Amazon Linux, Ubuntu Server oder RHEL ausgeführt wird, und das IAM-Instance-Profil anzuhängen, das Sie in erstellt haben. Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances Beispiel:

```
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**
>
> Dieser Befehl erstellt eine Standardsicherheitsgruppe für die Amazon EC2 EC2-Instance, die den Zugriff auf mehrere Ports ermöglicht, einschließlich des uneingeschränkten Zugriffs für SSH über Port 22 und alternativ HTTP über Port 80. Als bewährte Methode empfehlen wir, den Zugriff nur auf die SSH- und HTTP-Ports zu beschränken. CodeDeploy erfordert keinen uneingeschränkten Portzugriff und erfordert keinen HTTP-Portzugriff. Weitere Informationen finden Sie unter Tipps zur Sicherung Ihrer Amazon EC2 EC2-Instance.

Für Windows Server

Um den run-instances Befehl aufzurufen, um eine Amazon EC2 EC2-Instance zu starten, auf der Windows Server ausgeführt wird, und das IAM-Instance-Profil anzuhängenSchritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances, in dem Sie erstellt haben, und den Namen der Sicherheitsgruppe anzugeben, die Sie in Schritt 1 erstellt haben. Beispiel:

```
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
```

Diese Befehle starten eine einzelne Amazon EC2 EC2-Instance mit dem angegebenen AMI, dem angegebenen key pair und dem angegebenen Instance-Typ mit dem angegebenen IAM-Instance-Profil und führen das angegebene Skript beim Start aus.

3. Beachten Sie den Wert von `InstanceID` in der Ausgabe. Wenn Sie diesen Wert vergessen, können Sie ihn später abrufen, indem Sie den describe-instances Befehl für das Amazon EC2 EC2-Instance-Schlüsselpaar aufrufen.

```
aws ec2 describe-instances --filters "Name=key-name,Values=keyName" --query
 "Reservations[*].Instances[*].[InstanceId]" --output text
```

Verwenden Sie die Instance-ID, um den create-tags Befehl aufzurufen, der die Amazon EC2 EC2-Instance kennzeichnet, sodass sie später während einer Bereitstellung gefunden werden

CodeDeploy kann. Im folgenden Beispiel ist das Tag benannt**`CodeDeployDemo`**, aber Sie können jedes beliebige Amazon EC2 EC2-Instance-Tag angeben.

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=CodeDeployDemo
```

Sie können einer Instance gleichzeitig mehrere Tags zuordnen. Beispiel:

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=testInstance
 Key=Region,Value=West Key=Environment,Value=Beta
```

Um zu überprüfen, ob die Amazon EC2 EC2-Instance gestartet wurde und alle Prüfungen bestanden hat, verwenden Sie die Instance-ID, um den describe-instance-status Befehl aufzurufen.

```
aws ec2 describe-instance-status --instance-ids instance-id --query
 "InstanceStatuses[*].InstanceStatus.[Status]" --output text
```

Wenn die Instance gestartet wurde und alle Prüfungen bestanden wurden, wird ok in der Ausgabe angezeigt.

Installieren Sie den Agenten CodeDeploy

Der CodeDeploy Agent muss auf Ihrer Amazon EC2 EC2-Instance installiert sein, bevor Sie ihn in CodeDeploy Bereitstellungen verwenden können. Weitere Informationen finden Sie unter Installieren Sie den CodeDeploy Agenten.

> **Note**
>
> Sie können die automatische Installation und Updates des CodeDeploy Agenten konfigurieren, wenn Sie Ihre Bereitstellungsgruppe in der Konsole erstellen.

Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy (CloudFormation Vorlage)

Sie können unsere CloudFormation Vorlage verwenden, um schnell eine Amazon EC2 EC2-Instance zu starten, auf der Amazon Linux oder Windows Server ausgeführt wird. Sie können die AWS CLI,

die CodeDeploy Konsole oder die AWS APIs verwenden, um die Instance mit der Vorlage zu starten. Zusätzlich zum Starten der Instance übernimmt die Vorlage folgende Aufgaben:

- Weist an CloudFormation , der Instanz die Erlaubnis zu erteilen, an CodeDeploy Bereitstellungen teilzunehmen.
- Markiert die Instanz, sodass sie während einer Bereitstellung gefunden werden CodeDeploy kann.
- Installiert den CodeDeploy Agenten und führt ihn auf der Instanz aus.

Sie müssen unsere CloudFormation nicht verwenden, um eine Amazon EC2 EC2-Instance einzurichten. Alternativen finden Sie unter Arbeiten mit Instanzen für CodeDeploy.

Wir stellen keine CloudFormation Vorlage für Amazon EC2 EC2-Instances bereit, auf denen Ubuntu Server oder Red Hat Enterprise Linux (RHEL) ausgeführt werden.

Themen

- Bevor Sie beginnen
- Starten Sie eine Amazon EC2 EC2-Instance mit dem CloudFormation Vorlage (Konsole)
- Starten Sie eine Amazon EC2 EC2-Instance mit dem CloudFormation Vorlage (AWS CLI)

Bevor Sie beginnen

Bevor Sie die CloudFormation Vorlage zum Starten von Amazon EC2 EC2-Instances verwenden können, stellen Sie sicher, dass Sie die folgenden Schritte ausführen.

1. Stellen Sie sicher, dass Sie einen Administratorbenutzer erstellt haben, wie unter beschriebenSchritt 1: Einrichtung. Double-check Stellen Sie sicher, dass der Benutzer über die folgenden Mindestberechtigungen verfügt, und fügen Sie alle hinzu, die nicht vorhanden sind:

   - Wolkenbildung: *
   - codedeploy:*
   - ec2:*
   - ich bin: AddRoleToInstanceProfile
   - ich bin: CreateInstanceProfile
   - ich bin: CreateRole
   - ich bin: DeleteInstanceProfile
   - ich bin: DeleteRole

- ich bin: DeleteRolePolicy
- ich bin: GetRole
- ich bin: DeleteRolePolicy
- ich bin: PutRolePolicy
- ich bin: RemoveRoleFromInstanceProfile

2. Stellen Sie sicher, dass Sie über ein Instance-Schlüsselpaar verfügen, um den SSH-Zugriff auf die Amazon EC2-Instance, auf der Amazon Linux ausgeführt wird, oder den RDP-Zugriff auf die Instance, auf der Windows Server ausgeführt wird, zu ermöglichen.

   Um einen Schlüsselpaarnamen zu finden, öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2. Wählen Sie im Navigationsbereich unter Network & Security (Netzwerk und Sicherheit) Key Pairs (Schlüsselpaare) aus und notieren Sie den Schlüsselpaarnamen in der Liste.

   Informationen zum Generieren eines neuen key pair finden Sie unter Erstellen Ihres key pair mit Amazon EC2. Stellen Sie sicher, dass das key pair in einer der unter Region aufgeführten Regionen und Endpunkte in Allgemeine AWS-Referenzerstellt wurde. Ansonsten können Sie das Instance-Schlüsselpaar nicht mit CodeDeploy verwenden.

Starten Sie eine Amazon EC2 EC2-Instance mit dem CloudFormation Vorlage (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

   > **Important**
   >
   > Melden Sie sich AWS-Managementkonsole mit demselben Konto an, das Sie verwendet habenErste Schritte mit CodeDeploy. Wählen Sie in der Navigationsleiste in der Regionsauswahl eine der Regionen aus, die unter Region und Endpunkte in aufgeführt sind. Allgemeine AWS-Referenz CodeDeploy unterstützt nur diese Regionen.

2. Wählen Sie Stack erstellen aus.
3. Wählen Sie unter Vorlage auswählen die Option Geben Sie eine Amazon S3 S3-Vorlagen-URL an. Geben Sie in das Feld den Speicherort der CloudFormation Vorlage für Ihre Region ein und wählen Sie dann Weiter aus.

Region	Speicherort der CloudFormation Vorlage
Region USA Ost (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Region USA Ost (Nord-Virginia)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region USA West (Nordkalifornien)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region USA West (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Kanada (Zentral)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Europa (Irland)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Europa (London)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Speicherort der CloudFormation Vorlage
Region Europa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Europa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Israel (Tel Aviv)	`http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Asien-Pazifik (Hongkong)-Region	`http://s3-ap-east-1.amazonaws.com/aws-codedeploy-ap-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Asia Pacific (Tokyo) Region	`http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Asia Pacific (Seoul) Region	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Speicherort der CloudFormation Vorlage
Region Asien-Pazifik (Singapur)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Asien-Pazifik (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Asien-Pazifik (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Asien-Pazifik (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Südamerika (São Paulo)	`aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`

4. Geben Sie in das Feld Stack name (Stack-Name) einen Namen ein (z. B. **`CodeDeployDemoStack`**).
5. Geben Sie im Feld Parameters (Parameter) Folgendes ein und wählen Sie dann Next (Weiter) aus.

   - Geben Sie für InstanceCountdie Anzahl der Instances ein, die Sie starten möchten. (Wir empfehlen, den Standardwert 1 beizubehalten.)

- Geben Sie für den Instance-Typ ein InstanceType, den Sie starten möchten (oder behalten Sie den Standardwert t1.micro bei).
- Geben Sie für KeyPairNameden Namen des Instanzschlüsselpaars key pair. Geben Sie nur den Namen des Schlüsselpaars ein, nicht die Dateierweiterung des Schlüsselpaars.
- Geben Sie im OperatingSystemFeld ein, **`Windows`** um Instanzen zu starten, auf denen Windows Server ausgeführt wird (oder behalten Sie die Standardeinstellung Linux bei).
- Geben Sie für SSHLocation den IP-Adressbereich ein, der für die Verbindung mit der Instanz über SSH oder RDP verwendet werden soll (oder behalten Sie den Standardwert 0.0.0 bei. 0/0).

> **Important**
>
> Die Standardeinstellung von **`0.0.0.0/0`** wird nur zu Demonstrationszwecken bereitgestellt. CodeDeploy erfordert nicht, dass Amazon EC2 EC2-Instances uneingeschränkten Zugriff auf Ports haben. Als bewährte Methode empfehlen wir, Zugriff auf SSH- (und HTTP)-Ports einzuschränken. Weitere Informationen finden Sie unter Tipps zur Sicherung Ihrer Amazon EC2 EC2-Instance.

- Geben Sie den Instance-Tag-Schlüssel ein TagKey, mit dem die Instances während der Bereitstellung identifiziert CodeDeploy werden sollen (oder behalten Sie die Standardeinstellung Name bei).
- Geben Sie für den Instanz-Tag-Wert ein TagValue, mit dem die Instances während der Bereitstellung identifiziert CodeDeploy werden sollen (oder behalten Sie den Standardwert von bei CodeDeployDemo).

6. Lassen Sie auf der Seite Options (Optionen) die Optionsfelder leer, und wählen Sie Next (Weiter).

> **Important**
>
> CloudFormation Tags unterscheiden sich von CodeDeploy Tags. CloudFormation verwendet Tags, um die Verwaltung Ihrer Infrastruktur zu vereinfachen. CodeDeploy verwendet Tags, um Amazon EC2 EC2-Instances zu identifizieren. Sie haben CodeDeploy-Tags auf der Seite Specify Parameters (Parameter angeben) angegeben.

7. Wählen Sie auf der Seite Überprüfen unter Funktionen das Feld Ich bestätige, dass CloudFormation möglicherweise IAM-Ressourcen erstellt werden, und wählen Sie dann Erstellen aus.

   Nachdem CloudFormation der Stack erstellt und die Amazon EC2 EC2-Instances gestartet wurden, wird CREATE_COMPLETE in der Spalte Status in der CloudFormation Konsole angezeigt. Dieser Vorgang kann einige Minuten dauern.

Informationen zur Überprüfung, ob der CodeDeploy Agent auf den Amazon EC2 EC2-Instances ausgeführt wird, finden Sie unterVerwaltung des CodeDeploy Agentenbetriebs, und fahren Sie dann fort mit. Erstellen Sie eine Anwendung mit CodeDeploy

Starten Sie eine Amazon EC2 EC2-Instance mit dem CloudFormation Vorlage (AWS CLI)

1. Verwenden Sie unsere CloudFormation Vorlage in einem Aufruf des create-stack Befehls. Dieser Stack startet eine neue Amazon EC2 EC2-Instance, auf der der CodeDeploy Agent installiert ist.

   So starten Sie eine Amazon EC2 EC2-Instance, auf der Amazon Linux ausgeführt wird:

   ```
   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
   ```

   So starten Sie eine Amazon EC2 EC2-Instance, auf der Windows Server ausgeführt wird:

   ```
   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*ist der Name des Instanzschlüsselpaars. Geben Sie nur den Namen des Schlüsselpaars ein, nicht die Dateierweiterung des Schlüsselpaars.

*template-url*ist der Speicherort der CloudFormation Vorlage für Ihre Region:

Region	Speicherort der CloudFormation Vorlage
Region USA Ost (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Region USA Ost (Nord-Virginia)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region USA West (Nordkalifornien)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region USA West (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Kanada (Zentral)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Speicherort der CloudFormation Vorlage
Region Europa (Irland)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Europa (London)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Europa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Europa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Israel (Tel Aviv)	`http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Asien-Pazifik (Hongkong)-Region	`http://s3-ap-east-1.amazonaws.com/aws-codedeploy-ap-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Speicherort der CloudFormation Vorlage
Asia Pacific (Tokyo) Region	`http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Asia Pacific (Seoul) Region	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Asien-Pazifik (Singapur)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Asien-Pazifik (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Asien-Pazifik (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`
Region Asien-Pazifik (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Speicherort der CloudFormation Vorlage
Region Südamerika (São Paulo)	`aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`

Dieser Befehl erstellt einen CloudFormation Stack mit dem Namen**`CodeDeployDemoStack`**, der die CloudFormation Vorlage im angegebenen Amazon S3 S3-Bucket verwendet. Die Amazon EC2 EC2-Instance basiert auf dem Instance-Typ t1.micro, Sie können jedoch jeden beliebigen Typ verwenden. Sie ist mit dem Wert **`CodeDeployDemo`** getaggt, aber Sie können sie mit jedem beliebigen Wert taggen. Das angegebene Instance-Schlüsselpaar wurde darauf angewendet.

2. Rufen Sie den describe-stacks Befehl auf, um zu überprüfen, ob der angegebene CloudFormation Stack erfolgreich erstellt **`CodeDeployDemoStack`** wurde:

```
aws cloudformation describe-stacks --stack-name CodeDeployDemoStack --query
 "Stacks[0].StackStatus" --output text
```

Fahren Sie erst fort, wenn der Wert `CREATE_COMPLETE` zurückgegeben wurde.

Informationen zur Überprüfung, ob der CodeDeploy Agent auf der Amazon EC2 EC2-Instance ausgeführt wird, finden Sie unterVerwaltung des CodeDeploy Agentenbetriebs, und fahren Sie dann fort mit. Erstellen Sie eine Anwendung mit CodeDeploy

Konfigurieren Sie eine Amazon EC2 EC2-Instance, mit der Sie arbeiten können CodeDeploy

Diese Anweisungen zeigen Ihnen, wie Sie eine Amazon EC2 EC2-Instance, auf der Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) oder Windows Server ausgeführt wird, für die Verwendung in CodeDeploy Bereitstellungen konfigurieren.

> **Note**
>
> Wenn Sie keine Amazon EC2 EC2-Instance haben, können Sie die CloudFormation Vorlage verwenden, um eine Instance zu starten, auf der Amazon Linux oder Windows Server ausgeführt wird. Wir bieten keine Vorlage für Ubuntu Server oder RHEL an.

Schritt 1: Stellen Sie sicher, dass ein IAM-Instance-Profil an Ihre Amazon EC2 EC2-Instance angehängt ist

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances die Option Instances.
3. Navigieren Sie zu Ihrer Amazon EC2 EC2-Instance und wählen Sie sie in der Liste aus.
4. Notieren Sie sich im Detailbereich auf der Registerkarte Beschreibung den Wert im Feld IAM-Rolle und fahren Sie dann mit dem nächsten Abschnitt fort.

   Wenn das Feld leer ist, können Sie der Instance ein IAM-Instanzprofil anhängen. Weitere Informationen finden Sie unter Eine IAM-Rolle an eine Instance anhängen.

Schritt 2: Stellen Sie sicher, dass das angehängte IAM-Instanzprofil über die richtigen Zugriffsberechtigungen verfügt

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen.
3. Navigieren Sie zu dem IAM-Rollennamen, den Sie sich in Schritt 4 des vorherigen Abschnitts notiert haben, und wählen Sie ihn aus.

   > **Note**
   >
   > Wenn Sie die durch die CloudFormation Vorlage generierte Servicerolle anstelle einer Servicerolle verwenden möchten, die Sie anhand der Anweisungen unter erstellt habenSchritt 2: Erstellen Sie eine Servicerolle für CodeDeploy, beachten Sie Folgendes: In einigen Versionen unserer CloudFormation Vorlage entspricht der Anzeigename des generierten und an die Amazon EC2 EC2-Instances angehängten IAM-Instance-Profils nicht dem Anzeigenamen in der IAM-Konsole. Beispielsweise könnte das IAM-

Instance-Profil den Anzeigenamen haben`CodeDeploySampleStack-expnyi6-InstanceRoleInstanceProfile-IK8J8A9123EX`, während das IAM-Instance-Profil in der IAM-Konsole den Anzeigenamen haben könnte. `CodeDeploySampleStack-expnyi6-InstanceRole-C5P33V1L64EX`
Um Ihnen bei der Identifizierung des Instanzprofils in der IAM-Konsole zu helfen, sehen Sie, dass das Präfix von für beide `CodeDeploySampleStack-expnyi6-InstanceRole` identisch ist. Informationen darüber, warum diese Anzeigenamen unterschiedlich sein können, finden Sie unter Instanzprofile.

4. Wählen Sie die Registerkarte Trust Relationships. Wenn es in Trusted Entities keinen Eintrag mit der Aufschrift The identity provider (s) ec2.amazonaws.com gibt, können Sie diese Amazon EC2 EC2-Instance nicht verwenden. Stoppen und erstellen Sie eine Amazon EC2 EC2-Instance mithilfe der Informationen inArbeiten mit Instanzen für CodeDeploy.

   Wenn es einen Eintrag mit der Aufschrift The identity provider (s) ec2.amazonaws.com gibt und Sie Ihre Anwendungen nur in GitHub Repositorys speichern, fahren Sie mit fort. Schritt 3: Markieren Sie die Amazon EC2 EC2-Instance

   Wenn es einen Eintrag mit der Aufschrift The identity provider (s) ec2.amazonaws.com gibt und Sie Ihre Anwendungen in Amazon S3 S3-Buckets speichern, wählen Sie den Tab Berechtigungen.

5. Wenn es eine Richtlinie im Bereich Permissions policies (Berechtigungsrichtlinien) gibt, erweitern Sie die Richtlinie und wählen dann Edit policy (Richtlinie bearbeiten).

6. Wählen Sie den Tab JSON. Wenn Sie Ihre Anwendungen in Amazon S3 S3-Buckets speichern, stellen Sie sicher`"s3:Get*"`, dass sie in der Liste der angegebenen Aktionen aufgeführt `"s3:List*"` sind.

   Dies kann etwa wie folgt aussehen:

```
{"Statement":[{"Resource":"*","Action":[
  ... Some actions may already be listed here ...
  "s3:Get*","s3:List*"
  ... Some more actions may already be listed here ...
  ],"Effect":"Allow"}]}
```

   Oder es kann wie folgt aussehen:

JSON

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

Wenn `"s3:Get*"` und `"s3:List*"` nicht in der Liste der angegebenen Aktionen stehen, verwenden Sie Edit (Bearbeiten), um sie hinzuzufügen, und wählen Sie dann Save (Speichern). (Wenn weder `"s3:Get*"` noch `"s3:List*"` die letzte Aktion in der Liste ist, fügen Sie ein Komma direkt nach der Aktion ein, sodass das Richtliniendokument validiert wird.)

> **Note**
>
> Wir empfehlen, diese Richtlinie nur auf die Amazon S3 S3-Buckets zu beschränken, auf die Ihre Amazon EC2 EC2-Instances zugreifen müssen. Stellen Sie sicher, dass Sie Zugriff auf die Amazon S3 S3-Buckets gewähren, die den CodeDeploy Agenten enthalten. Andernfalls kann ein Fehler auftreten, wenn der CodeDeploy Agent auf den Instances installiert oder aktualisiert wird. Um dem IAM-Instance-Profil nur Zugriff auf einige CodeDeploy Resource Kit-Buckets in Amazon S3 zu gewähren, verwenden Sie die folgende Richtlinie, entfernen Sie jedoch die Zeilen für Buckets, auf die Sie den Zugriff verhindern möchten:
>
> ```
> {
>   "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/*"
      ]
    }
  ]
}
```

Schritt 3: Markieren Sie die Amazon EC2 EC2-Instance

Anweisungen dazu, wie Sie die Amazon EC2 EC2-Instance taggen, sodass sie während einer Bereitstellung gefunden werden CodeDeploy kann, finden Sie unter Arbeiten mit Tags in der Konsole. Kehren Sie dann zu dieser Seite zurück.

> **Note**
>
> Sie können die Amazon EC2 EC2-Instance mit einem beliebigen Schlüssel und Wert kennzeichnen. Denken Sie daran, diesen Schlüssel und den Wert bei der Bereitstellung anzugeben.

Schritt 4: Installieren Sie AWS CodeDeploy Agent auf der Amazon EC2 EC2-Instance

Anweisungen zur Installation des CodeDeploy Agenten auf der Amazon EC2 EC2-Instance und zur Überprüfung, ob er ausgeführt wirdVerwaltung des CodeDeploy Agentenbetriebs, finden Sie unter und fahren Sie dann mit fortErstellen Sie eine Anwendung mit CodeDeploy.

Arbeiten mit lokalen Instanzen für CodeDeploy

Eine lokale Instance ist jedes physische Gerät, bei dem es sich nicht um eine Amazon EC2 EC2-Instance handelt, auf dem der CodeDeploy Agent ausgeführt und eine Verbindung zu öffentlichen AWS Service-Endpunkten hergestellt werden kann.

Die Bereitstellung einer CodeDeploy Anwendungsversion für eine lokale Instance umfasst zwei wichtige Schritte:

- Schritt 1 — Konfigurieren Sie jede lokale Instanz, registrieren Sie sie bei und CodeDeploy taggen Sie sie anschließend.
- Schritt 2 — Stellen Sie Anwendungsrevisionen auf der lokalen Instanz bereit.

> **Note**
>
> Informationen dazu, wie Sie mit dem Erstellen und Bereitstellen einer Beispielanwendungsrevision auf einer korrekt konfigurierten und registrierten lokalen Instance experimentieren, finden Sie unter Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat

Enterprise Linux). Informationen zu lokalen Instanzen und deren Verwendung finden Sie unter CodeDeploy. Working with On-Premises Instances

Wenn Sie nicht mehr möchten, dass eine lokale Instanz in Bereitstellungen verwendet wird, können Sie die lokalen Instanz-Tags aus den Bereitstellungsgruppen entfernen. Sie können die Tags der lokalen Instance auch von der Instance entfernen. Sie können auch explizit die Registrierung einer lokalen Instance aufheben, sodass sie in Bereitstellungen nicht mehr verwendet werden kann. Weitere Informationen finden Sie unter Verwaltung von Vorgängen für lokale Instanzen in CodeDeploy.

Die Anweisungen in diesem Abschnitt zeigen Ihnen, wie Sie eine lokale Instanz konfigurieren und sie anschließend registrieren und taggen, CodeDeploy damit sie in Bereitstellungen verwendet werden kann. In diesem Abschnitt wird auch beschrieben, wie Sie CodeDeploy Informationen über lokale Instances abrufen und die Registrierung einer lokalen Instance aufheben können, wenn Sie nicht mehr planen, auf ihr zu implementieren.

Themen

- Voraussetzungen für die Konfiguration einer lokalen Instanz
- Registrieren Sie eine lokale Instanz mit CodeDeploy
- Verwaltung von Vorgängen für lokale Instanzen in CodeDeploy

Voraussetzungen für die Konfiguration einer lokalen Instanz

Die folgenden Voraussetzungen müssen erfüllt sein, bevor Sie eine lokale Instanz registrieren können.

Important

Wenn Sie den Befehl register-on-premises-instance verwenden und die temporären Anmeldeinformationen, die mit () generiert werden, regelmäßig aktualisieren, gelten weitere Voraussetzungen. AWS -Security-Token-Service AWS STS Weitere Informationen finden Sie unter Voraussetzungen für die ARN-Registrierung der IAM-Sitzung.

Anforderungen an Speichergeräte

Auf dem Gerät, das Sie vorbereiten, registrieren und als lokale Instanz kennzeichnen möchten, muss ein unterstütztes Betriebssystem ausgeführt werden. CodeDeploy Eine Liste finden Sie hier: Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy.

Wenn Ihr Betriebssystem nicht unterstützt wird, steht Ihnen der CodeDeploy Agent als Open-Source-Lösung zur Verfügung, sodass Sie ihn an Ihre Bedürfnisse anpassen können. Weitere Informationen finden Sie im CodeDeploy Agenten-Repository unter GitHub.

Ausgehende Kommunikation

Die lokale Instanz muss in der Lage sein, eine Verbindung zu Endpunkten des öffentlichen AWS Dienstes herzustellen, mit denen sie kommunizieren kann. CodeDeploy

Der CodeDeploy Agent kommuniziert ausgehend über HTTPS über Port 443.

Administrative Kontrolle

Das lokale Konto oder das Netzwerkkonto, das auf der lokalen Instanz zur Konfiguration der lokalen Instanz verwendet wird, muss entweder als `sudo` oder `root` (für Ubuntu Server) oder als Administrator (für Windows Server) ausgeführt werden können.

IAM-Berechtigungen

Der IAM-Identität, die Sie für die Registrierung der lokalen Instanz verwenden, müssen Berechtigungen zum Abschließen der Registrierung (und zum Abmelden der lokalen Instanz, falls erforderlich) erteilt werden.

Stellen Sie zusätzlich zu der unter beschriebenen Richtlinie sicherSchritt 3: Beschränken Sie die CodeDeploy Benutzerberechtigungen, dass der aufrufenden IAM-Identität die folgende zusätzliche Richtlinie beigefügt ist.

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": "*"
    }
  ]
}
```

Informationen zum Zuweisen von IAM-Richtlinien finden Sie unter Verwalten von IAM-Richtlinien.

Registrieren Sie eine lokale Instanz mit CodeDeploy

Um eine lokale Instance zu registrieren, müssen Sie eine IAM-Identität zur Authentifizierung Ihrer Anforderungen verwenden. Sie können zwischen den folgenden IAM-Identitäts- und Registrierungsmethoden auswählen:

- Verwenden Sie einen IAM-Rollen-ARN zum Authentifizieren von Anforderungen.
  - Verwenden Sie den Befehl register-on-premises-instance und regelmäßig aktualisierte temporäre Anmeldeinformationen, die mit () generiert werden, um die AWS -Security-Token-Service meisten Registrierungsoptionen manuell zu konfigurieren.AWS STS Diese Option bietet die höchste Sicherheitsstufe, da die Authentifizierung mithilfe eines temporären Tokens erfolgt, bei dem das Timeout überschritten wird und das regelmäßig aktualisiert werden muss. Diese Option wird für Produktionsbereitstellungen jeder Größe empfohlen. Weitere Informationen finden Sie unter Verwenden Sie den Befehl register-on-premises-instance (IAM Session ARN), um eine lokale Instanz zu registrieren.
- (Nicht empfohlen) Verwenden Sie einen IAM-Benutzer-ARN, um Anfragen zu authentifizieren.
  - Verwenden Sie den Befehl register für den automatisiertesten Registrierungsprozess. Diese Option sollte nur für Bereitstellungen außerhalb der Produktion verwendet werden, bei denen die Sicherheit weniger wichtig ist. Diese Option ist weniger sicher, da sie statische (permanente) Anmeldeinformationen für die Authentifizierung verwendet. Diese Option eignet sich gut für die Registrierung einer einzelnen lokalen Instanz. Weitere Informationen finden Sie unter Verwenden Sie den Befehl register (IAM-Benutzer-ARN), um eine lokale Instance zu registrieren.
  - Verwenden Sie den Befehl register-on-premises-instance, um die meisten Registrierungsoptionen manuell zu konfigurieren. Für die Registrierung einer kleinen Anzahl

lokaler Instances geeignet. Weitere Informationen finden Sie unter Verwenden Sie den Befehl register-on-premises-instance (IAM-Benutzer-ARN), um eine lokale Instance zu registrieren.

Themen

- Verwenden Sie den Befehl register-on-premises-instance (IAM Session ARN), um eine lokale Instanz zu registrieren
- Verwenden Sie den Befehl register (IAM-Benutzer-ARN), um eine lokale Instance zu registrieren
- Verwenden Sie den Befehl register-on-premises-instance (IAM-Benutzer-ARN), um eine lokale Instance zu registrieren

Verwenden Sie den Befehl register-on-premises-instance (IAM Session ARN), um eine lokale Instanz zu registrieren

Um maximale Kontrolle über die Authentifizierung und Registrierung Ihrer lokalen Instances zu erhalten, können Sie den Befehl register-on-premises-instance und regelmäßig aktualisierte temporäre Anmeldeinformationen verwenden, die mit () generiert werden. AWS -Security-Token-Service AWS STS Eine statische IAM-Rolle für die Instanz übernimmt die Rolle dieser aktualisierten Anmeldeinformationen für die Durchführung von Bereitstellungsvorgängen. AWS STS CodeDeploy

Diese Methode ist besonders nützlich, wenn Sie eine große Anzahl von Instances registrieren müssen. Sie ermöglicht es Ihnen, den Registrierungsprozess mit zu automatisieren. CodeDeploy Sie können Ihr eigenes Identitäts- und Authentifizierungssystem verwenden, um lokale Instanzen zu authentifizieren und die Anmeldeinformationen für IAM-Sitzungen vom Dienst an die Instanzen zu verteilen, mit denen sie verwendet werden können. CodeDeploy

> **Note**
>
> Alternativ können Sie einen gemeinsam genutzten IAM-Benutzer verwenden, der auf alle lokalen Instanzen verteilt ist, um die AWS STS AssumeRoleAPI aufzurufen, um Sitzungsanmeldeinformationen für lokale Instanzen abzurufen. Diese Methode ist weniger sicher und wird nicht für Produktions- oder geschäftskritische Umgebungen empfohlen.

Verwenden Sie die Informationen in den folgenden Themen, um eine lokale Instanz mithilfe temporärer Sicherheitsanmeldeinformationen zu konfigurieren, die mit generiert wurden. AWS STS

Themen

- Voraussetzungen für die ARN-Registrierung der IAM-Sitzung
- Schritt 1: Erstellen Sie die IAM-Rolle, die lokale Instanzen übernehmen sollen
- Schritt 2: Generieren Sie temporäre Anmeldeinformationen für eine einzelne Instanz mit AWS STS
- Schritt 3: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu
- Schritt 4: Bereiten Sie eine lokale Instanz für Bereitstellungen vor CodeDeploy
- Schritt 5: Registrieren Sie die lokale Instanz bei CodeDeploy
- Schritt 6: Kennzeichnen Sie die lokale Instanz
- Schritt 7: Stellen Sie Anwendungsrevisionen auf der lokalen Instanz bereit
- Schritt 8: Verfolgen Sie die Bereitstellungen auf der lokalen Instanz

Voraussetzungen für die ARN-Registrierung der IAM-Sitzung

Zusätzlich zu den unter Voraussetzungen für die Konfiguration einer lokalen Instanz aufgeführten Voraussetzungen müssen die folgenden zusätzlichen Anforderungen erfüllt sein:

IAM-Berechtigungen

Der IAM-Identität, die Sie zur Registrierung einer lokalen Instanz verwenden, müssen Berechtigungen zur Ausführung von Vorgängen erteilt werden. CodeDeploy Stellen Sie sicher, dass die verwaltete AWSCodeDeployFullAccess-Richtlinie der IAM-Identität zugeordnet ist. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter AWS Verwaltete Richtlinien.

System zum Aktualisieren temporärer Anmeldeinformationen

Wenn Sie einen IAM-Sitzungs-ARN zum Registrieren lokaler Instances verwenden, müssen Sie über ein System zum regelmäßigen Aktualisieren der temporären Anmeldeinformationen verfügen. Temporäre Anmeldeinformationen laufen nach einer Stunde ab, oder früher, wenn bei der Generierung der Anmeldeinformationen ein kürzerer Zeitraum angegeben wird. Es gibt zwei Methoden zum Aktualisieren der Anmeldeinformationen:

- Methode 1: Verwenden Sie das Identitäts- und Authentifizierungssystem in Ihrem Unternehmensnetzwerk mit einem CRON-Skript, das regelmäßig das Identitäts- und Authentifizierungssystem abfragt und die neuesten Sitzungsanmeldeinformationen auf die Instance kopiert. Auf diese Weise können Sie Ihre Authentifizierungs- und Identitätsstruktur integrieren, AWS ohne Änderungen am CodeDeploy Agenten oder Dienst vornehmen zu müssen, um die in Ihrer Organisation verwendeten Authentifizierungstypen zu unterstützen.

- Methode 2: Führen Sie regelmäßig einen CRON-Job auf der Instanz aus, um die AWS STS AssumeRoleAktion aufzurufen und die Sitzungsanmeldedaten in eine Datei zu schreiben, auf die der CodeDeploy Agent zugreifen kann. Bei dieser Methode müssen Sie weiterhin einen IAM-Benutzer verwenden und Anmeldeinformationen auf die lokale Instance kopieren, jedoch können Sie dieselben IAM-Benutzer- und -Anmeldeinformationen in der gesamten Flotte lokaler Instances wiederverwenden.

Informationen zum Erstellen und Arbeiten mit AWS STS Anmeldeinformationen finden Sie unter AWS -Security-Token-Service API-Referenz und Verwenden temporärer Sicherheitsanmeldedaten, um Zugriff auf AWS Ressourcen anzufordern.

Schritt 1: Erstellen Sie die IAM-Rolle, die lokale Instanzen übernehmen sollen

Sie können die AWS CLI oder die IAM-Konsole verwenden, um eine IAM-Rolle zu erstellen, die von Ihren lokalen Instances zur Authentifizierung und Interaktion verwendet wird. CodeDeploy

Sie müssen nur eine IAM-Rolle erstellen. Jede Ihrer lokalen Instances kann diese Rolle übernehmen, um die temporären Sicherheitsanmeldeinformationen abzurufen, die die dieser Rolle erteilten Berechtigungen gewähren.

Für die Rolle, die Sie erstellen, sind die folgenden Berechtigungen erforderlich, um auf die Dateien zuzugreifen, die für die Installation des Agenten erforderlich sind: CodeDeploy

JSON

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

Wir empfehlen, dass Sie diese Richtlinie nur auf die Amazon S3 S3-Buckets beschränken, auf die Ihre lokale Instance zugreifen muss. Wenn Sie diese Richtlinie einschränken, stellen Sie sicher, dass Sie Zugriff auf die Amazon S3 S3-Buckets gewähren, die den CodeDeploy Agenten enthalten. Andernfalls kann ein Fehler auftreten, wenn der CodeDeploy Agent auf der lokalen Instance installiert oder aktualisiert wird. Informationen zur Steuerung des Zugriffs auf Amazon S3 S3-Buckets finden Sie unter Zugriffsberechtigungen für Ihre Amazon S3 S3-Ressourcen verwalten.

So erstellen Sie die IAM-Rolle

1. Rufen Sie den Befehl create-role mit der Option `--role-name` auf, um einen Namen für die IAM-Rolle anzugeben (z. B. `CodeDeployInstanceRole`), und mit der Option `--assume-role-policy-document`, um die Berechtigungen zu erteilen.

   Wenn Sie die IAM-Rolle für diese Instance erstellen, können Sie ihr den Namen `CodeDeployInstanceRole` geben und die erforderlichen Berechtigungen in eine Datei mit dem Namen `CodeDeployRolePolicy.json` einschließen:

   ```
   aws iam create-role --role-name CodeDeployInstanceRole --assume-role-policy-
   document file://CodeDeployRolePolicy.json
   ```

2. Notieren Sie den Wert des ARN-Felds aus der Ausgabe des Aufrufs für den Befehl create-role. Beispiel:

   ```
   arn:aws:iam::123456789012:role/CodeDeployInstanceRole
   ```

   Sie benötigen den Rollen-ARN, wenn Sie die AWS STS AssumeRoleAPI verwenden, um kurzfristige Anmeldeinformationen für jede Instanz zu generieren.

   Weitere Informationen zum Erstellen von IAM-Rollen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Dienst im IAM-Benutzerhandbuch.

   Informationen zum Zuweisen von Berechtigungen zu einer vorhandenen Rolle finden Sie unter put-role-policy in der Befehlsreferenz.AWS CLI

Schritt 2: Generieren Sie temporäre Anmeldeinformationen für eine einzelne Instanz mit AWS STS

Bevor Sie die temporären Anmeldeinformationen generieren, die für die Registrierung einer lokalen Instance verwendet werden, müssen Sie die IAM-Identität (Benutzer oder Rolle) erstellen oder auswählen, für die Sie die temporären Anmeldeinformationen generieren. Die Berechtigung

`sts:AssumeRole` muss für diese IAM-Identität in die Richtlinieneinstellungen eingeschlossen werden.

Informationen zum Erteilen von `sts:AssumeRole` Berechtigungen für eine IAM-Identität finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Dienst und. AssumeRole

Es gibt zwei Möglichkeiten zum Generieren der temporären Anmeldeinformationen:

- Verwenden Sie den Befehl assume-role mit dem. AWS CLI Beispiel:

  ```
  aws sts assume-role --role-arn arn:aws:iam::12345ACCOUNT:role/role-arn --role-
  session-name session-name
  ```

  Wobei Folgendes gilt:

  - *12345ACCOUNT*ist die 12-stellige Kontonummer für Ihre Organisation.
  - *role-arn*ist der ARN der zu übernehmenden Rolle, den Sie generiert habenSchritt 1: Erstellen Sie die IAM-Rolle, die lokale Instanzen übernehmen sollen.
  - *session-name*ist der Name, den Sie der Rollensitzung geben möchten, die Sie gerade erstellen.

  > **Note**
  >
  > Wenn Sie ein CRON-Skript verwenden, das das Identitäts- und Authentifizierungssystem regelmäßig abfragt und die neuesten Sitzungsanmeldedaten in die Instanz kopiert (Methode 1 zum Aktualisieren temporärer Anmeldeinformationen, beschrieben unterVoraussetzungen für die ARN-Registrierung der IAM-Sitzung), können Sie stattdessen jedes unterstützte AWS SDK zum Aufrufen verwenden. AssumeRole

- Verwenden Sie ein von bereitgestelltes Tool. AWS

  Das Tool aws-codedeploy-session-helper generiert AWS STS Anmeldeinformationen und schreibt sie in eine Datei, die Sie auf der Instanz platzieren. Dieses Tool eignet sich am besten für Methode 2 zum Aktualisieren von temporären Anmeldeinformationen, die unter Voraussetzungen für die ARN-Registrierung der IAM-Sitzung beschrieben wird. Bei dieser Methode wird das Tool aws-codedeploy-session-helper auf jeder Instanz platziert und führt den Befehl mit den Berechtigungen eines IAM-Benutzers aus. Jede Instanz verwendet die Anmeldeinformationen desselben IAM-Benutzers in Verbindung mit diesem Tool.

Weitere Informationen finden Sie im GitHub aws-codedeploy-session-helper-Repository.

> **Note**
>
> Nach dem Erstellen der Anmeldeinformationen für die IAM-Sitzung platzieren Sie sie irgendwo auf der lokalen Instance. Im nächsten Schritt konfigurieren Sie den CodeDeploy Agenten so, dass er auf die Anmeldeinformationen an diesem Speicherort zugreift.

Bevor Sie fortfahren, stellen Sie sicher, dass das System, das Sie zur regelmäßigen Aktualisierung der temporären Anmeldeinformationen verwenden, vorhanden ist. Wenn die temporären Anmeldeinformationen nicht aktualisiert werden, schlagen Bereitstellungen auf der lokalen Instance fehl. Weitere Informationen finden Sie im Abschnitt „System to refresh temporary credentials“ unter Voraussetzungen für die ARN-Registrierung der IAM-Sitzung.

Schritt 3: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu

Fügen Sie einer lokalen Instance eine Konfigurationsdatei mithilfe von Root- oder Administratorberechtigungen. Diese Konfigurationsdatei wird verwendet, um die IAM-Anmeldeinformationen und die AWS Zielregion zu deklarieren, für die verwendet werden soll. CodeDeploy Die Datei muss an einem bestimmten Speicherort in der lokalen Instance abgelegt werden. Die Datei muss den ARN der temporären IAM-Sitzung, die zugehörige geheime Schlüssel-ID und den geheimen Zugriffsschlüssel sowie die AWS Zielregion enthalten.

Hinzufügen eine Konfigurationsdatei

1. Erstellen Sie eine Datei mit dem Namen `codedeploy.onpremises.yml` (für eine lokale Ubuntu Server- oder RHEL-Instanz) oder `conf.onpremises.yml` (für eine lokale Windows Server-Instanz) am folgenden Speicherort auf der lokalen Instanz:
   - Für Ubuntu Server: `/etc/codedeploy-agent/conf`
   - Für Windows Server: `C:\ProgramData\Amazon\CodeDeploy`
2. Verwenden Sie einen Texteditor, um der neu erstellten `codedeploy.onpremises.yml` Datei (Linux) oder `conf.onpremises.yml` Datei (Windows) die folgenden Informationen hinzuzufügen:

```
---
iam_session_arn: iam-session-arn
```

```
aws_credentials_file: credentials-file
region: supported-region
```

Wobei Folgendes gilt:

- *iam-session-arn*ist der ARN der IAM-Sitzung, den Sie notiert haben. Schritt 2: Generieren Sie temporäre Anmeldeinformationen für eine einzelne Instanz mit AWS STS
- *credentials-file*ist der Speicherort der Anmeldeinformationsdatei für den temporären Sitzungs-ARN, wie unter angegebenSchritt 2: Generieren Sie temporäre Anmeldeinformationen für eine einzelne Instanz mit AWS STS.
- *supported-region*ist eine der Regionen, die CodeDeploy unterstützt werden, wie unter Region und Endgeräte in Allgemeine AWS-Referenzaufgeführt.

Schritt 4: Bereiten Sie eine lokale Instanz für Bereitstellungen vor CodeDeploy

Installieren und konfigurieren Sie AWS CLI

Installieren und konfigurieren Sie die AWS CLI On-Premises-Instanz. (Der AWS CLI wird verwendet, um den CodeDeploy Agenten herunterzuladen und auf der lokalen Instanz zu installieren.)

1. Folgen Sie den Anweisungen AWS CLI unter Getting Setup with the im AWS Command Line Interface Benutzerhandbuch, um den auf der lokalen Instanz AWS CLI zu installieren.

   > **Note**
   >
   > CodeDeploy Befehle für die Arbeit mit lokalen Instanzen wurden in Version 1.7.19 von verfügbar. AWS CLI Wenn Sie AWS CLI bereits eine Version von installiert haben, können Sie deren Version überprüfen, indem Sie anrufen. aws --version

2. Um die AWS CLI On-Premises-Instanz zu konfigurieren, folgen Sie den Anweisungen unter Konfiguration der AWS CLI im AWS Command Line Interface Benutzerhandbuch.

   > **Important**
   >
   > Achten Sie bei der Konfiguration der AWS CLI (z. B. durch Aufrufen des aws configure Befehls) darauf, die geheime Schlüssel-ID und den geheimen Zugriffsschlüssel eines IAM-Benutzers anzugeben, der mindestens über die unter beschriebenen Berechtigungen verfügt. Voraussetzungen für die ARN-Registrierung der IAM-Sitzung

Festlegen der Umgebungsvariablen AWS_REGION (nur Ubuntu Server und RHEL)

Wenn Sie Ubuntu Server oder RHEL nicht auf Ihrer lokalen Instanz ausführen, überspringen Sie diesen Schritt und fahren Sie direkt mit „Agent installieren“ fort. CodeDeploy

Installieren Sie den CodeDeploy Agenten auf einer lokalen Ubuntu-Server- oder RHEL-Instanz und ermöglichen Sie der Instanz, den CodeDeploy Agenten zu aktualisieren, sobald eine neue Version verfügbar ist. Dazu setzen Sie die `AWS_REGION` Umgebungsvariable auf der Instanz auf den Bezeichner einer der Regionen, die von unterstützt werden. CodeDeploy Es wird empfohlen, den Wert auf die Region festzulegen, in der sich Ihre CodeDeploy Anwendungen, Bereitstellungsgruppen und Anwendungsrevisionen befinden (z. B.`us-west-2`). Eine Liste der Regionen finden Sie unter Region und Endpunkte in der. Allgemeine AWS-Referenz

Rufen Sie zum Festlegen der Umgebungsvariable Folgendes vom Terminal ab:

```
export AWS_REGION=supported-region
```

Wo *`supported-region`* ist der Region-Identifier (zum Beispiel`us-west-2`).

Installieren Sie den CodeDeploy Agenten

- Folgen Sie für eine lokale Ubuntu-Server-Instanz den Anweisungen unter Installieren Sie den CodeDeploy Agenten für Ubuntu Server und kehren Sie dann zu dieser Seite zurück.
- Folgen Sie für eine lokale RHEL-Instanz den Anweisungen unter und kehren Sie dann zu dieser Seite zurück. Installieren Sie den CodeDeploy Agenten für Amazon Linux oder RHEL
- Folgen Sie für eine lokale Windows Server-Instanz den Anweisungen unter und kehren Sie dann zu dieser Seite zurück. Installieren Sie den CodeDeploy Agenten für Windows Server

Schritt 5: Registrieren Sie die lokale Instanz bei CodeDeploy

Bei diesen Anweisungen gehen wir davon aus, dass Sie die Registrierung des lokalen Instance von der lokalen Instance aus durchführen. Sie können eine lokale Instanz von einem separaten Gerät oder einer Instanz aus registrieren, auf der die Instanz AWS CLI installiert und konfiguriert ist.

Verwenden Sie den AWS CLI , um die lokale Instanz zu registrieren, CodeDeploy damit sie in Bereitstellungen verwendet werden kann.

Bevor Sie den verwenden können AWS CLI, benötigen Sie den ARN der temporären Sitzungsanmeldedaten, die Sie in erstellt habenSchritt 3: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu. Zum Beispiel für eine als `AssetTag12010298EX` identifizierte Instance:

```
arn:sts:iam::123456789012:assumed-role/CodeDeployInstanceRole/AssetTag12010298EX
```

Rufen Sie den Befehl register-on-premises-instance auf und geben Sie Folgendes an:

- Ein eindeutiger Bezeichner für die lokale Instance (mit der Option `--instance-name`).

> **Important**
>
> Um die lokale Instance identifizieren zu können, insbesondere für Debugging-Zwecke, empfehlen wir dringend, dass Sie einen Namen verwenden, der einige eindeutige Merkmale der lokalen Instance enthält (z. B. der session-name der STS-Anmeldeinformationen und die Seriennummer oder eine eindeutige interne Asset-ID, falls zutreffend). Wenn Sie eine MAC-Adresse als Namen angeben, beachten Sie, dass MAC-Adressen unzulässige Zeichen enthalten, z. B. einen CodeDeploy Doppelpunkt (:). Eine Liste der zulässigen Zeichen finden Sie unter CodeDeploy Kontingente.

- Den ARN der IAM-Sitzung, den Sie zur Authentifizierung mehrerer lokaler Instances in Schritt 1: Erstellen Sie die IAM-Rolle, die lokale Instanzen übernehmen sollen eingerichtet haben.

Beispiel:

```
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
```

Wobei Folgendes gilt:

- *`name-of-instance`*ist der Name, den Sie verwenden, um die lokale Instanz zu identifizieren, z. B. `AssetTag12010298EX`
- *`account-id`*ist die 12-stellige Konto-ID für Ihre Organisation, z. B. `111222333444`
- *`role-to-assume`*ist der Name der IAM-Rolle, die Sie für die Instance erstellt haben, z. B. `CodeDeployInstanceRole`
- *`session-name`*ist der Name der Sitzungsrolle, in Schritt 2: Generieren Sie temporäre Anmeldeinformationen für eine einzelne Instanz mit AWS STS der Sie angegeben haben.

Schritt 6: Kennzeichnen Sie die lokale Instanz

Sie können entweder die Konsole AWS CLI oder die CodeDeploy Konsole verwenden, um die lokale Instanz zu taggen. (CodeDeployverwendet lokale Instanz-Tags, um die Bereitstellungsziele während einer Bereitstellung zu identifizieren.)

So markieren Sie eine lokale Instance (CLI):

- Rufen Sie den Befehl add-tags-to-on-premises-instances auf und geben Sie Folgendes an:
  - Der eindeutige Bezeichner für die lokale Instance (mit der Option `--instance-names`).
  - Der Name des Tag-Schlüssels und des Tag-Werts der lokalen Instance, die Sie verwenden möchten (mit der Option `--tags`). Sie müssen sowohl einen Namen als auch einen Wert angeben. CodeDeploy erlaubt keine lokalen Instanz-Tags, die nur Werte enthalten.

    Beispiel:

    ```
    aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
     --tags Key=Name,Value=CodeDeployDemo-OnPrem
    ```

So markieren Sie eine lokale Instance (Konsole):

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie On-premises Instanzen aus.
3. Klicken Sie in der Liste der lokalen Instances auf den Namen der lokalen Instance, die Sie markieren möchten.
4. Wählen Sie in der Liste der Tags den gewünschten Tag-Schlüssel und -Wert aus oder geben Sie ihn ein. Nachdem Sie den Tag-Schlüssel und den Tag-Wert eingegeben haben, wird eine weitere Zeile angezeigt. Sie können das für bis zu 10 Tags wiederholen. Klicken Sie zum Entfernen eines Tags auf Remove (Entfernen).

5. Wählen Sie nach dem Hinzufügen der Tags die Option Update Tags.

Schritt 7: Stellen Sie Anwendungsrevisionen auf der lokalen Instanz bereit

Sie können nun auf den registrierten und mit einem Tag markierten lokalen Instances Anwendungsrevisionen bereitstellen.

Sie stellen Anwendungsrevisionen auf lokale Instances auf ähnliche Weise bereit, wie Sie Anwendungsrevisionen für Amazon EC2 EC2-Instances bereitstellen. Detaillierte Anweisungen finden Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy. Diese Anweisungen enthalten einen Link zu den Voraussetzungen, wie dem Erstellen einer Anwendung, Erstellen einer Bereitstellungsgruppe und Vorbereiten einer Anwendungsrevision. Wenn Sie eine einfache Beispielanwendungsrevision bereitstellen möchten, können Sie die unter Schritt 2: Erstellen Sie eine Version der Beispielanwendung im Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat Enterprise Linux) beschriebene Revision verwenden.

> **Important**
>
> Wenn Sie eine CodeDeploy Servicerolle im Rahmen der Erstellung einer Bereitstellungsgruppe für lokale Instances wiederverwenden, müssen Sie die Richtlinienerklärung der Servicerolle in diesen `Action` Teil aufnehmen`Tag:get*`. Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.

Schritt 8: Verfolgen Sie die Bereitstellungen auf der lokalen Instanz

Nachdem Sie eine Anwendungsrevision auf einer registrierten und markierten lokalen Instance bereitgestellt haben, können Sie den Fortschritt der Bereitstellung nachverfolgen.

Sie verfolgen Bereitstellungen für lokale Instances auf ähnliche Weise wie die Nachverfolgung von Bereitstellungen auf Amazon EC2 EC2-Instances. Detaillierte Anweisungen finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen.

Verwenden Sie den Befehl register (IAM-Benutzer-ARN), um eine lokale Instance zu registrieren

> **Important**
>
> Die Registrierung einer Instance mithilfe eines IAM-Benutzers wird nicht empfohlen, da für die Authentifizierung statische (permanente) Anmeldeinformationen verwendet werden. Zur Verbesserung der Sicherheit empfehlen wir, eine Instance mit temporären Anmeldeinformationen für die Authentifizierung zu registrieren. Weitere Informationen finden Sie unter Verwenden Sie den Befehl register-on-premises-instance (IAM Session ARN), um eine lokale Instanz zu registrieren.

> **Important**
>
> Stellen Sie sicher, dass Sie über einen Plan verfügen, um die Zugriffsschlüssel (permanente Anmeldeinformationen) des IAM-Benutzers zu rotieren. Weitere Informationen finden Sie unter Rotieren von Zugriffsschlüsseln.

In diesem Abschnitt wird beschrieben, wie Sie eine lokale Instance konfigurieren und registrieren und sie mit CodeDeploy mit dem geringsten Aufwand mit Tags versehen. Der Befehl register ist besonders nützlich, wenn Sie mit einer einzigen oder einer kleinen Anzahl von lokalen Instances arbeiten. Sie können den register Befehl nur verwenden, wenn Sie einen IAM-Benutzer-ARN zur Authentifizierung einer Instance verwenden. Sie können den register Befehl nicht mit einem IAM-Sitzungs-ARN zur Authentifizierung verwenden.

Wenn Sie den register Befehl verwenden, können Sie Folgendes CodeDeploy tun lassen:

- Erstellen Sie einen IAM-Benutzer AWS Identity and Access Management für die lokale Instance, falls Sie keinen mit dem Befehl angeben.
- Speichern Sie die Anmeldeinformationen des IAM-Benutzers in einer lokalen Instanzkonfigurationsdatei.
- Registrieren Sie die lokale Instanz bei. CodeDeploy
- Fügen Sie Tags zur lokalen Instance hinzu, indem Sie diese als Teil des Befehls angeben.

> **Note**
>
> Der Befehl register-on-premises-instance ist eine Alternative zum Befehl register. https://docs.aws.amazon.com/cli/latest/reference/deploy/register.html Sie verwenden den Befehl register-on-premises-instance, wenn Sie eine Instance vor Ort konfigurieren möchten und sie bei CodeDeploy eigenständig registrieren und mit Tags versehen möchten. Der register-on-premises-instance Befehl bietet Ihnen auch die Möglichkeit, anstelle eines IAM-Benutzer-ARN einen IAM-Sitzungs-ARN zur Registrierung von Instances zu verwenden. Dieser Ansatz bietet einen großen Vorteil, wenn Sie große Serverfarmen für lokale Instances haben. Insbesondere können Sie einen einzelnen IAM-Sitzungs-ARN verwenden, um mehrere Instanzen zu authentifizieren, anstatt für jede lokale Instance einzeln einen IAM-Benutzer erstellen zu müssen. Weitere Informationen erhalten Sie unter Verwenden Sie den Befehl register-on-premises-instance (IAM-Benutzer-ARN), um eine lokale Instance zu registrieren und Verwenden Sie den Befehl register-on-premises-instance (IAM Session ARN), um eine lokale Instanz zu registrieren.

Themen

- Schritt 1: Installieren und konfigurieren Sie AWS CLI auf der lokalen Instanz
- Schritt 2: Rufen Sie den Befehl register auf
- Schritt 3: Rufen Sie den Installationsbefehl auf
- Schritt 4: Stellen Sie Anwendungsrevisionen auf der lokalen Instanz bereit
- Schritt 5: Verfolgen Sie die Bereitstellungen auf der lokalen Instanz

Schritt 1: Installieren und konfigurieren Sie AWS CLI auf der lokalen Instanz

1. Installieren Sie die AWS CLI auf der lokalen Instanz. Folgen Sie den Anweisungen unter Erste Schritte mit dem AWS CLI im AWS Command Line Interface Benutzerhandbuch.

   > **Note**
   >
   > CodeDeploy Befehle für die Arbeit mit lokalen Instanzen sind in AWS CLI Version 1.7.19 und höher verfügbar. Wenn Sie das AWS CLI bereits installiert haben, rufen Sie an, aws --version um die Version zu überprüfen.

2. Konfigurieren Sie die AWS CLI auf der lokalen Instanz. Folgen Sie den Anweisungen unter Konfiguration des AWS CLI AWS Command Line Interface Benutzerleitfadens.

> **Important**
>
> Achten Sie bei der Konfiguration AWS CLI (z. B. durch Aufrufen des aws configure Befehls) darauf, die geheime Schlüssel-ID und den geheimen Zugriffsschlüssel eines IAM-Benutzers anzugeben, der zusätzlich zu den unter angegebenen Berechtigungen mindestens über die folgenden AWS Zugriffsberechtigungen verfügt. Voraussetzungen für die Konfiguration einer lokalen Instanz Dadurch ist es möglich, den CodeDeploy Agenten auf die lokale Instanz herunterzuladen und zu installieren. Die Zugriffsberechtigungen können etwa wie folgt aussehen:
>
> 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**
>
> Wenn Sie versuchen, auf einen der zuvor angezeigten Amazon S3-Buckets zuzugreifen, die Fehler „Zugriff verweigert“ sehen, versuchen Sie beispielsweise, den `/*` Teil des Ressourcen-ARN des Buckets wegzulassen. `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/*"
      ]
    }
  ]
}
```

Schritt 2: Rufen Sie den Befehl register auf

Bei diesem Schritt gehen wir davon aus, dass Sie die Registrierung des lokalen Instance von der lokalen Instance aus durchführen. Sie können eine lokale Instanz auch von einem separaten Gerät oder einer separaten Instanz aus registrieren, auf der die Instanz AWS CLI installiert und konfiguriert ist, wie im vorherigen Schritt beschrieben.

Verwenden Sie den AWS CLI , um den Befehl register aufzurufen, und geben Sie dabei Folgendes an:

- Ein Name, der die lokale Instanz eindeutig identifiziert CodeDeploy (mit der `--instance-name` Option).

  > **Important**
  >
  > Um die lokale Instance später identifizieren zu können, insbesondere für Debugging-Zwecke, empfehlen wir dringend, dass Sie einen Namen verwenden, die einige eindeutige Merkmale der lokalen Instance enthält (z. B. die Seriennummer oder eine eindeutige interne Asset-ID, falls zutreffend). Wenn Sie eine MAC-Adresse für einen Namen angeben, beachten Sie, dass MAC-Adressen Zeichen enthalten, die CodeDeploy nicht zulässig sind, wie z. B. Doppelpunkt (`:`). Eine Liste der zulässigen Zeichen finden Sie unter CodeDeploy Kontingente.

- Optional der ARN eines vorhandenen IAM-Benutzers, den Sie dieser lokalen Instance zuordnen möchten (mit der `--iam-user-arn` Option). Um den ARN eines IAM-Benutzers abzurufen, rufen

Sie den Befehl get-user auf, oder wählen Sie den IAM-Benutzernamen im Abschnitt Benutzer der IAM-Konsole aus und suchen Sie dann im Abschnitt Zusammenfassung nach dem Benutzer-ARN-Wert. Wenn diese Option nicht angegeben ist, CodeDeploy wird in Ihrem Namen in Ihrem AWS Konto ein IAM-Benutzer erstellt und dieser der lokalen Instance zugeordnet.

> **Important**
>
> Wenn Sie die Option `--iam-user-arn` festlegen, müssen Sie die lokale Instance-Konfigurationsdatei manuell erstellen, wie in Schritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu beschrieben.
> Sie können nur einen IAM-Benutzer mit nur einer lokalen Instance verknüpfen. Der Versuch, einen einzelnen IAM-Benutzer mehreren lokalen Instances zuzuordnen, kann zu Fehlern, fehlgeschlagenen Bereitstellungen auf diesen lokalen Instances oder zu Bereitstellungen auf lokalen Instances führen, die in einem permanenten Ausstehen-Status feststecken.

- Optional eine Reihe von lokalen Instance-Tags (mit der `--tags` Option), die zur Identifizierung der Amazon EC2-Instances verwendet CodeDeploy werden, für die die Bereitstellung verwendet werden soll. Geben Sie jeden Tag mit `Key=`*`tag-key`*`,Value=`*`tag-value`* an (z. B. `Key=Name,Value=Beta Key=Name,Value=WestRegion`). Wenn diese Option nicht angegeben ist, werden keine Tags registriert. Um Tags später zu registrieren, rufen Sie den Befehl https://docs.aws.amazon.com/cli/latest/reference/deploy/add-tags-to-on-premises-instances.html add-tags-to-on-premises-instances auf.
- Optional die AWS Region, in der die lokale Instanz registriert wird (mit der Option). CodeDeploy `--region` Dies muss eine der unterstützten Regionen sein, die unter Region aufgeführt sind, und Endpunkte in Allgemeine AWS-Referenz (z. B.`us-west-2`). Wenn diese Option nicht angegeben ist, wird die AWS Standardregion verwendet, die dem anrufenden IAM-Benutzer zugeordnet ist.

Beispiel:

```
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
```

Der Befehl register hat folgende Auswirkungen:

1. Wenn kein vorhandener IAM-Benutzer angegeben ist, wird ein IAM-Benutzer erstellt, ihm die erforderlichen Berechtigungen zugewiesen und ein entsprechender geheimer Schlüssel und ein geheimer Zugriffsschlüssel generiert. Die lokale Instance verwendet diesen IAM-Benutzer und seine Berechtigungen und Anmeldeinformationen, um sich zu authentifizieren und mit ihm zu interagieren. CodeDeploy
2. Registriert die lokale Instanz bei. CodeDeploy
3. Falls angegeben, ordnet es CodeDeploy den Tags, die mit der `--tags` Option angegeben sind, dem registrierten lokalen Instanznamen zu.
4. Wenn ein IAM-Benutzer erstellt wurde, erstellt außerdem die erforderliche Konfigurationsdatei in demselben Verzeichnis, aus dem der register Befehl aufgerufen wurde.

Wenn dieser Befehl Fehler erkennt, wird eine Fehlermeldung angezeigt, in der beschrieben wird, wie Sie die verbleibenden Schritte manuell durchführen können. Andernfalls wird eine Erfolgsmeldung angezeigt, die beschreibt, wie Sie den Befehl install aufrufen, wie im nächsten Schritt beschrieben.

Schritt 3: Rufen Sie den Installationsbefehl auf

Verwenden Sie von der lokalen Instanz aus den, AWS CLI um den https://docs.aws.amazon.com/cli/latest/reference/deploy/install.html Installationsbefehl aufzurufen, und geben Sie dabei Folgendes an:

- Der Pfad zur Konfigurationsdatei (mit der Option `--config-file`).
- Optional, ob die Konfigurationsdatei ersetzt werden soll, die bereits auf der lokalen Instance vorhanden ist (mit der `--override-config` Option). Wenn nicht angegeben, wird die vorhandene Konfigurationsdatei nicht ersetzt.
- Optional die AWS Region, in der die lokale Instanz registriert wird CodeDeploy (mit der `--region` Option). Dies muss eine der unterstützten Regionen sein, die unter Region aufgeführt sind, und Endpunkte in Allgemeine AWS-Referenz (z. B.`us-west-2`). Wenn diese Option nicht angegeben ist, wird die AWS Standardregion verwendet, die dem anrufenden IAM-Benutzer zugeordnet ist.
- Optional ein benutzerdefinierter Speicherort, von dem aus der CodeDeploy Agent installiert werden soll (mit der `--agent-installer` Option). Diese Option ist nützlich, um eine benutzerdefinierte Version des CodeDeploy Agenten zu installieren, die CodeDeploy nicht offiziell unterstützt wird (z. B. eine benutzerdefinierte Version, die auf dem CodeDeploy Agenten-Repository in basiert GitHub). Der Wert muss der Pfad zu einem Amazon S3-Bucket sein, der entweder Folgendes enthält:
  - Ein CodeDeploy Agenteninstallationsskript (für Linux- oder Unix-based Betriebssysteme, ähnlich der Installationsdatei im CodeDeploy Agenten-Repository in GitHub).

- Eine CodeDeploy Agenteninstallationspaketdatei (.msi) (für Windows-based Betriebssysteme).

Wenn diese Option nicht angegeben ist, CodeDeploy wird nach besten Kräften versucht, von ihrem eigenen Standort aus eine offiziell unterstützte Version des CodeDeploy Agenten zu installieren, die mit dem Betriebssystem auf der lokalen Instanz kompatibel ist.

Beispiel:

```
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
```

Der Befehl install hat folgende Auswirkungen:

1. Überprüft, ob es sich bei der lokalen Instance um eine Amazon EC2-Instance handelt. Wenn dies der Fall ist, wird eine Fehlermeldung angezeigt.
2. Kopiert die Konfigurationsdatei für lokale Instances vom angegebenen Speicherort auf der Instance an den Speicherort, an dem der CodeDeploy Agent sie voraussichtlich finden wird, vorausgesetzt, die Datei befindet sich nicht bereits an diesem Speicherort.

   Für Ubuntu Server und Red Hat Enterprise Linux (RHEL) ist `/etc/codedeploy-agent/conf` dies/. `codedeploy.onpremises.yml`

   Für Windows Server ist dies`C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.

   Wenn die Option `--override-config` angegeben wurde, wird die Datei erstellt oder überschrieben.
3. Installiert den CodeDeploy Agenten auf der lokalen Instanz und startet ihn dann.

Schritt 4: Stellen Sie Anwendungsrevisionen auf der lokalen Instanz bereit

Sie können nun auf den registrierten und mit einem Tag markierten lokalen Instances Anwendungsrevisionen bereitstellen.

Sie stellen Anwendungsrevisionen auf lokalen Instances auf eine Weise bereit, die der Bereitstellung von Anwendungsrevisionen auf Amazon EC2-Instances ähnelt. Detaillierte Anweisungen finden Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy. Diese Anweisungen bieten einen Link zu den Voraussetzungen, wie dem Erstellen einer Anwendung, Erstellen einer

Bereitstellungsgruppe und Vorbereiten einer Anwendungsrevision. Wenn Sie eine einfache Beispielanwendungsrevision bereitstellen möchten, können Sie die unter Schritt 2: Erstellen Sie eine Version der Beispielanwendung im Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat Enterprise Linux) beschriebene Revision verwenden.

> **Important**
>
> Wenn Sie im Rahmen der Erstellung einer Bereitstellungsgruppe, die auf lokale Instances abzielt, eine vorhandene CodeDeploy Servicerolle wiederverwenden, müssen Sie dem `Action` Teil der `Tag:get*` Richtlinienerklärung der Servicerolle beifügen. Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.

Schritt 5: Verfolgen Sie die Bereitstellungen auf der lokalen Instanz

Nachdem Sie eine Anwendungsrevision auf einer registrierten und markierten lokalen Instance bereitgestellt haben, können Sie den Fortschritt der Bereitstellung nachverfolgen.

Sie verfolgen Bereitstellungen auf lokalen Instances auf eine Weise, die der Verfolgung von Bereitstellungen auf Amazon EC2-Instances ähnelt. Detaillierte Anweisungen finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen.

Weitere Optionen finden Sie unter Verwaltung von Vorgängen für lokale Instanzen in CodeDeploy.

Verwenden Sie den Befehl register-on-premises-instance (IAM-Benutzer-ARN), um eine lokale Instance zu registrieren

> **Important**
>
> Die Registrierung einer Instanz mithilfe eines IAM-Benutzers wird nicht empfohlen, da für die Authentifizierung statische (permanente) Anmeldeinformationen verwendet werden. Zur Verbesserung der Sicherheit empfehlen wir, eine Instance mit temporären Anmeldeinformationen für die Authentifizierung zu registrieren. Weitere Informationen finden Sie unter Verwenden Sie den Befehl register-on-premises-instance (IAM Session ARN), um eine lokale Instanz zu registrieren.

> **Important**
>
> Stellen Sie sicher, dass Sie über einen Plan verfügen, um die Zugriffsschlüssel (permanente Anmeldeinformationen) des IAM-Benutzers zu rotieren. Weitere Informationen finden Sie unter Rotieren von Zugriffsschlüsseln.

Befolgen Sie diese Anweisungen zum Konfigurieren einer lokalen Instance. Registrieren Sie sie und markieren Sie sie weitestgehend selbst mit CodeDeploy , wobei Sie die statischen IAM-Benutzeranmeldeinformationen für die Authentifizierung verwenden.

Themen

- Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance
- Schritt 2: Weisen Sie dem IAM-Benutzer Berechtigungen zu
- Schritt 3: Rufen Sie die IAM-Benutzeranmeldedaten ab
- Schritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu
- Schritt 5: Installieren und konfigurieren Sie den AWS CLI
- Schritt 6: Stellen Sie die Umgebungsvariable AWS_REGION ein (nur Ubuntu Server und RHEL)
- Schritt 7: Installieren Sie den CodeDeploy Agenten
- Schritt 8: Registrieren Sie die lokale Instanz bei CodeDeploy
- Schritt 9: Taggen Sie die lokale Instanz
- Schritt 10: Stellen Sie Anwendungsrevisionen auf der lokalen Instanz bereit
- Schritt 11: Verfolgen Sie die Bereitstellungen auf der lokalen Instanz

Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance

Erstellen Sie einen IAM-Benutzer, den die lokale Instanz zur Authentifizierung und Interaktion verwendet. CodeDeploy

> **Important**
>
> Sie müssen für jede teilnehmende lokale Instance einen separaten IAM-Benutzer erstellen. Wenn Sie versuchen, einen einzelnen IAM-Benutzer für mehrere lokale Instances wiederzuverwenden, können Sie diese lokalen Instances möglicherweise nicht erfolgreich

registrieren oder taggen. CodeDeploy Bereitstellungen auf solchen lokalen Instances können in einem dauerhaften schwebenden Zustand hängenbleiben oder ganz fehlschlagen.

Wir empfehlen, dem IAM-Benutzer einen Namen zuzuweisen, der seinen Zweck identifiziert, z. B. CodeDeployUser-OnPrem

Sie können die AWS CLI oder die IAM-Konsole verwenden, um einen IAM-Benutzer zu erstellen. Weitere Informationen finden Sie unter Einen IAM-Benutzer in Ihrem Konto erstellen. AWS

Important

Notieren Sie sich den für den Benutzer bereitgestellten Benutzer-ARN, unabhängig davon, ob Sie die IAM-Konsole AWS CLI oder die IAM-Konsole verwenden, um einen neuen IAM-Benutzer zu erstellen. Diese Informationen sind später in Schritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu und Schritt 8: Registrieren Sie die lokale Instanz bei CodeDeploy erforderlich.

Schritt 2: Weisen Sie dem IAM-Benutzer Berechtigungen zu

Wenn Ihre lokale Instance Anwendungsrevisionen aus Amazon S3-Buckets bereitstellt, müssen Sie dem IAM-Benutzer die Berechtigungen für die Interaktion mit diesen Buckets zuweisen. Sie können die AWS CLI oder die IAM-Konsole verwenden, um Berechtigungen zuzuweisen.

Note

Wenn Sie Anwendungsrevisionen nur aus GitHub Repositorys bereitstellen, überspringen Sie diesen Schritt und gehen Sie direkt zu. Schritt 3: Rufen Sie die IAM-Benutzeranmeldedaten ab (Sie benötigen weiterhin Informationen über den IAM-Benutzer, in dem Sie erstellt haben. Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance Es wird in späteren Schritten verwendet.)

So weisen Sie Berechtigungen (CLI) zu:

1. Erstellen Sie eine Datei mit den folgenden Richtlinieninhalten auf der Amazon EC2-Instance oder dem Gerät, mit dem Sie die AWS CLI aufrufen. Verwenden Sie einen Dateinamen wie etwa **`CodeDeploy-OnPrem-Permissions.json`** und speichern Sie die Datei.

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

Note

Wir empfehlen, dass Sie diese Richtlinie nur auf die Amazon S3-Buckets beschränken, auf die Ihre lokale Instance zugreifen muss. Wenn Sie diese Richtlinie einschränken, stellen Sie sicher, dass Sie auch Zugriff auf die Amazon S3-Buckets gewähren, die den Agenten enthalten. AWS CodeDeploy Andernfalls kann ein Fehler auftreten, wenn der CodeDeploy Agent auf der zugehörigen lokalen Instance installiert oder aktualisiert wird. Beispiel:

```
{
  "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. Rufen Sie den Befehl put-user-policy auf und geben Sie den Namen des IAM-Benutzers (mit der `--user-name` Option), einen Namen für die Richtlinie (mit der `--policy-name` Option) und den Pfad zum neu erstellten Richtliniendokument (mit der Option) an. `--policy-document` Wenn sich z. B. die Datei **`CodeDeploy-OnPrem-Permissions.json`** im selben Verzeichnis (Ordner) befindet, aus dem Sie diesen Befehl aufrufen:

> **Important**
>
> Achten Sie darauf, dass `file://` vor dem Dateinamen steht. Dies ist bei diesem Befehl erforderlich.

```
aws iam put-user-policy --user-name CodeDeployUser-OnPrem --policy-name CodeDeploy-OnPrem-Permissions --policy-document file://CodeDeploy-OnPrem-Permissions.json
```

So weisen Sie Berechtigungen (Konsole) zu:

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Richtlinien und dann Richtlinie erstellen. (Wenn die Schaltfläche Get Started (Erste Schritte) angezeigt wird, klicken Sie darauf und wählen Sie anschließend Create Policy (Richtlinie erstellen) aus.)
3. Klicken Sie neben Create Your Own Policy auf Select.
4. Geben Sie in das Feld Policy Name einen Namen für diese Richtlinie ein (z. B. **`CodeDeploy-OnPrem-Permissions`**).
5. Geben Sie im Feld Richtliniendokument den folgenden Berechtigungsausdruck ein oder fügen Sie ihn ein, der es ermöglicht AWS CodeDeploy , Anwendungsrevisionen aus jedem in der Richtlinie angegebenen Amazon S3-Bucket im Namen des IAM-Benutzers auf der lokalen Instance bereitzustellen:

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

6. Wählen Sie Richtlinie erstellen aus.
7. Klicken Sie im Navigationsbereich auf Users (Benutzer).
8. Suchen Sie in der Benutzerliste nach dem Namen des IAM-Benutzers, in dem Sie ihn erstellt haben, und wählen Sie ihn aus. Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance
9. Wählen Sie auf der Registerkarte Permissions unter Managed Policies die Option Attach Policy.
10. Wählen Sie die Richtlinie mit dem Namen **`CodeDeploy-OnPrem-Permissions`** aus, und klicken Sie dann auf Attach Policy (Richtlinie anfügen).

Schritt 3: Rufen Sie die IAM-Benutzeranmeldedaten ab

Rufen Sie die geheime Schlüssel-ID und den geheimen Zugriffsschlüssel für den IAM-Benutzer ab. Sie benötigen diese für Schritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu. Sie können die AWS CLI oder die IAM-Konsole verwenden, um die geheime Schlüssel-ID und den geheimen Zugriffsschlüssel abzurufen.

Note

Wenn Sie die geheime Schlüssel-ID und den geheimen Zugriffsschlüssel bereits haben, überspringen Sie diesen Schritt und gehen Sie direkt zu Schritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu über.
Benutzer benötigen programmgesteuerten Zugriff, wenn sie AWS außerhalb von interagieren möchten. AWS-Managementkonsole Die Art und Weise, wie programmgesteuerter Zugriff gewährt wird, hängt vom Benutzertyp ab, der zugreift. AWS
Um Benutzern programmgesteuerten Zugriff zu gewähren, wählen Sie eine der folgenden Optionen.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
IAM	(Empfohlen) Verwenden Sie Konsolenanmeldedaten als temporäre Anmeldeinformationen, um programmatische Anforderungen an die AWS SDKs AWS CLI oder APIs zu signieren. AWS	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu finden Sie im AWS Command Line Interface Benutzerhandbuch unter Anmeldung für AWS lokale Entwicklung. AWS CLI • Informationen zu AWS SDKs finden Sie unter Anmeldung für AWS lokale Entwicklung im

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
		Referenzhandbuch für AWS SDKs und Tools.
Mitarbeiteridentität (Benutzer, die in IAM Identity Center verwaltet werden)	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an die AWS SDKs oder AWS CLI APIs zu signieren . AWS	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu finden Sie AWS IAM Identity Center im AWS Command Line Interface Benutzerhandbuch unter Configuring AWS CLI to use. AWS CLI • Informationen zu AWS SDKs, Tools und AWS APIs finden Sie unter IAM Identity Center-Authentifizierung im Referenzhandbuch für AWS SDKs und Tools.
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an die AWS SDKs oder APIs AWS CLI zu signieren . AWS	Folgen Sie den Anweisungen unter Verwenden temporärer Anmeldeinformationen mit AWS Ressourcen im IAM-Benutzerhandbuch.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
IAM	(Nicht empfohlen) Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu finden Sie im AWS CLI Benutzerhandbuch unter Authentifizierung mit IAM-Benutzeranmeldeinformationen. AWS Command Line Interface • Informationen zu AWS SDKs und Tools finden Sie unter Authentifizieren mit langfristigen Anmeldeinformationen im Referenzhandbuch für AWS SDKs und Tools. • Informationen zu AWS APIs finden Sie unter Verwalten von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch.

So rufen Sie Anmeldeinformationen (CLI) ab:

1. Rufen Sie den Befehl list-access-keys auf, geben Sie den Namen des IAM-Benutzers (mit der `--user-name` Option) an und fragen Sie nur die Zugriffsschlüssel-IDs ab (mit den Optionen und). `--query` `--output` Beispiel:

```
aws iam list-access-keys --user-name CodeDeployUser-OnPrem --query
 "AccessKeyMetadata[*].AccessKeyId" --output text
```

2. Wenn in der Ausgabe keine Schlüssel oder nur Informationen zu einem Schlüssel in der Ausgabe erscheinen, rufen Sie den Befehl create-access-key auf und geben Sie den Namen des IAM-Benutzers an (mit der Option): `--user-name`

```
aws iam create-access-key --user-name CodeDeployUser-OnPrem
```

Notieren Sie den Wert der Felder `AccessKeyId` und `SecretAccessKey` aus der Ausgabe des Aufrufs des Befehls create-access-key. Sie benötigen diese Informationen in Schritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu.

> **Important**
>
> Das ist das einzige Mal, dass Sie Zugriff auf den geheimen Zugriffsschlüssel erhalten. Wenn Sie diesen geheimen Zugriffsschlüssel vergessen oder verlieren, müssen Sie einen neuen Zugriffsschlüssel erstellen, indem Sie die Schritte in Schritt 3: Rufen Sie die IAM-Benutzeranmeldedaten ab befolgen.

3. Wenn bereits zwei Zugriffsschlüssel aufgeführt sind, müssen Sie einen davon löschen, indem Sie den Befehl delete-access-key aufrufen und dabei den Namen des IAM-Benutzers (mit der Option) und die ID des zu löschenden Zugriffsschlüssels (mit der `--user-name` Option) angeben. `--access-key-id` Rufen Sie dann den Befehl create-access-key auf, wie weiter oben in diesem Schritt beschrieben. Es folgt ein Beispiel für einen Aufruf des Befehls delete-access-key:

```
aws iam delete-access-key --user-name CodeDeployUser-OnPrem --access-key-id access-key-ID
```

> **Important**
>
> Wenn Sie den delete-access-key Befehl aufrufen, um einen dieser Zugriffsschlüssel zu löschen, und eine lokale Instance diesen Zugriffsschlüssel bereits verwendet, wie unter beschriebenSchritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu, müssen Sie die Anweisungen unter Schritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu erneut befolgen, um eine andere Zugriffsschlüssel-ID

und einen anderen geheimen Zugriffsschlüssel anzugeben, die diesem IAM-Benutzer zugeordnet sind. Andernfalls bleiben Bereitstellungen auf dieser lokalen Instance in einem dauerhaften schwebenden Zustand hängen oder schlagen ganz fehl.

So rufen Sie Anmeldeinformationen (Konsole) ab:

1. a. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
   b. Wenn die Benutzerliste nicht angezeigt wird, klicken Sie im Navigationsbereich auf Users.
   c. Suchen Sie in der Benutzerliste nach dem Namen des IAM-Benutzers, in dem Sie ihn erstellt haben, und wählen Sie ihn aus. Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance
2. Wählen Sie auf der Registerkarte Security credentials, wenn kein oder nur ein Schlüssel aufgelistet ist, die Option Create access key.

   Wenn zwei Zugriffsschlüssel aufgelistet sind, müssen Sie einen davon löschen. Wählen Sie Delete neben einem der Zugriffsschlüssel, und wählen Sie dann Create access key.

   Important

   Wenn Sie neben einem dieser Zugriffsschlüssel auf Löschen klicken und eine lokale Instance diesen Zugriffsschlüssel bereits verwendet, wie unter beschrieben, müssen Sie die Anweisungen unter Schritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu erneut befolgenSchritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu, um eine andere Zugriffsschlüssel-ID und einen anderen geheimen Zugriffsschlüssel anzugeben, die diesem IAM-Benutzer zugeordnet sind. Andernfalls bleiben Bereitstellungen auf dieser lokalen Instance in einem dauerhaften schwebenden Zustand hängen oder schlagen ganz fehl.
3. Wählen Sie Show, und notieren Sie sich die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel. Diese Informationen sind für den nächsten Schritt erforderlich. Alternativ können Sie Download .csv file wählen, um eine Kopie der Zugriffsschlüssel-ID und des geheimen Zugriffsschlüssels zu speichern.

> **Important**
>
> Wenn Sie sich die Anmeldeinformationen nicht notieren oder herunterladen, ist es das einzige Mal, dass Sie Zugriff auf diesen geheimen Zugriffsschlüssel haben. Wenn Sie diesen geheimen Zugriffsschlüssel vergessen oder verlieren, müssen Sie einen neuen Zugriffsschlüssel erstellen, indem Sie die Schritte in Schritt 3: Rufen Sie die IAM-Benutzeranmeldedaten ab befolgen.

4. Wählen Sie Schließen, um zur Seite Benutzer > ***IAM User Name*** zurückzukehren.

Schritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu

Fügen Sie einer lokalen Instance eine Konfigurationsdatei mithilfe von Root- oder Administratorberechtigungen. Diese Konfigurationsdatei wird verwendet, um die IAM-Benutzeranmeldedaten und die AWS Zielregion, für die verwendet werden soll, zu deklarieren. CodeDeploy Die Datei muss an einem bestimmten Speicherort in der lokalen Instance abgelegt werden. Die Datei muss den ARN, die geheime Schlüssel-ID, den geheimen Zugriffsschlüssel und die AWS Zielregion des IAM-Benutzers enthalten. Die Datei muss in einem bestimmten Format vorliegen.

1. Erstellen Sie eine Datei mit dem Namen `codedeploy.onpremises.yml` (für eine lokale Ubuntu Server- oder RHEL-Instanz) oder `conf.onpremises.yml` (für eine lokale Windows Server-Instanz) am folgenden Speicherort auf der lokalen Instanz:

   - Für Ubuntu Server: `/etc/codedeploy-agent/conf`
   - Für Windows Server: `C:\ProgramData\Amazon\CodeDeploy`

2. Verwenden Sie einen Texteditor, um die folgenden Informationen der neu erstellten `codedeploy.onpremises.yml`- oder `conf.onpremises.yml`-Datei hinzuzufügen:

```
---
aws_access_key_id: secret-key-id
aws_secret_access_key: secret-access-key
iam_user_arn: iam-user-arn
region: supported-region
```

   Wobei Folgendes gilt:

- *secret-key-id*ist die geheime Schlüssel-ID des entsprechenden IAM-Benutzers, die Sie in Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance oder Schritt 3: Rufen Sie die IAM-Benutzeranmeldedaten ab notiert haben.
- *secret-access-key*ist der geheime Zugriffsschlüssel des entsprechenden IAM-Benutzers, den Sie in Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance oder notiert haben. Schritt 3: Rufen Sie die IAM-Benutzeranmeldedaten ab
- *iam-user-arn*ist der ARN des IAM-Benutzers, den Sie zuvor notiert haben. Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance
- *supported-region*ist die Kennung einer Region, die vom CodeDeploy Standort Ihrer CodeDeploy Anwendungen, Bereitstellungsgruppen und Anwendungsrevisionen unterstützt wird (z. B.`us-west-2`). Eine Liste der Regionen finden Sie unter Region und Endpunkte im. Allgemeine AWS-Referenz

> **Important**
>
> Wenn Sie neben einem der Zugriffsschlüssel unter Löschen ausgewählt haben und Ihre lokale Instance bereits die zugehörige Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel verwendet, müssen Sie den Anweisungen unter folgen, um eine andere Zugriffsschlüssel-ID und einen anderen geheimen Zugriffsschlüssel anzugeben, die diesem IAM-Benutzer zugeordnet sind. Schritt 3: Rufen Sie die IAM-Benutzeranmeldedaten ab Schritt 4: Fügen Sie der lokalen Instanz eine Konfigurationsdatei hinzu Andernfalls bleiben Bereitstellungen auf dieser lokalen Instance in einem dauerhaften schwebenden Zustand hängen oder schlagen ganz fehl.

Schritt 5: Installieren und konfigurieren Sie den AWS CLI

Installieren und konfigurieren Sie die AWS CLI auf der lokalen Instanz. (Der AWS CLI wird verwendetSchritt 7: Installieren Sie den CodeDeploy Agenten, um den CodeDeploy Agenten auf der lokalen Instanz herunterzuladen und zu installieren.)

1. Um die AWS CLI auf der lokalen Instance zu installieren, folgen Sie den Anweisungen unter Getting set up with the AWS CLI im AWS Command Line Interface Benutzerhandbuch.

> **Note**
>
> CodeDeploy Befehle für die Arbeit mit lokalen Instanzen wurden in Version 1.7.19 von verfügbar. AWS CLI Wenn Sie AWS CLI bereits eine Version von installiert haben, können Sie deren Version überprüfen, indem Sie anrufen. aws --version

2. Um die AWS CLI auf der lokalen Instanz zu konfigurieren, folgen Sie den Anweisungen unter Konfiguration der AWS CLI im AWS Command Line Interface Benutzerhandbuch.

> **Important**
>
> Achten Sie bei der Konfiguration AWS CLI (z. B. durch Aufrufen des aws configure Befehls) darauf, die Geheimschlüssel-ID und den geheimen Zugriffsschlüssel eines IAM-Benutzers anzugeben, der zusätzlich zu den in der angegebenen AWS Zugriffsberechtigungen mindestens über die folgenden Zugriffsberechtigungen verfügt. Voraussetzungen für die Konfiguration einer lokalen Instanz Auf diese Weise können Sie den CodeDeploy Agenten auf die lokale Instanz herunterladen und installieren:
>
> 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" : [
        "*"
      ]
    }
  ]
}
```

Diese Zugriffsberechtigungen können entweder dem IAM-Benutzer zugewiesen werden, in dem Sie erstellt haben, Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance oder einem anderen IAM-Benutzer. Um diese Berechtigungen einem IAM-Benutzer zuzuweisen, folgen Sie den Anweisungen unter und verwenden Sie diese Zugriffsberechtigungen anstelle der in diesem Schritt angegebenen. Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance

Schritt 6: Stellen Sie die Umgebungsvariable AWS_REGION ein (nur Ubuntu Server und RHEL)

Wenn Sie Ubuntu Server oder RHEL nicht auf Ihrer lokalen Instanz ausführen, überspringen Sie diesen Schritt und gehen Sie direkt zu. Schritt 7: Installieren Sie den CodeDeploy Agenten

Installieren Sie den CodeDeploy Agenten auf einer lokalen Ubuntu Server- oder RHEL-Instanz und ermöglichen Sie der Instanz, den CodeDeploy Agenten zu aktualisieren, sobald eine neue Version verfügbar ist. Dazu setzen Sie die `AWS_REGION` Umgebungsvariable auf der Instanz auf den Bezeichner einer der Regionen, die unterstützt werden von. CodeDeploy Wir empfehlen, den Wert auf die Region festzulegen, in der sich Ihre CodeDeploy Anwendungen, Bereitstellungsgruppen und Anwendungsrevisionen befinden (z. B.`us-west-2`). Eine Liste der Regionen finden Sie unter Region und Endpunkte im. Allgemeine AWS-Referenz

Rufen Sie zum Festlegen der Umgebungsvariable Folgendes vom Terminal ab:

```
export AWS_REGION=supported-region
```

Wo *`supported-region`* ist der Region-Identifier (z. B.`us-west-2`).

Schritt 7: Installieren Sie den CodeDeploy Agenten

Installieren Sie den CodeDeploy Agenten auf der lokalen Instanz:

- Folgen Sie für eine lokale Ubuntu Server-Instanz den Anweisungen unter und kehren Sie dann zu dieser Seite zurück. Installieren Sie den CodeDeploy Agenten für Ubuntu Server
- Folgen Sie für eine lokale RHEL-Instanz den Anweisungen unter und kehren Sie dann zu dieser Seite zurück. Installieren Sie den CodeDeploy Agenten für Amazon Linux oder RHEL
- Folgen Sie für eine lokale Windows Server-Instanz den Anweisungen unter und kehren Sie dann zu dieser Seite zurück. Installieren Sie den CodeDeploy Agenten für Windows Server

Schritt 8: Registrieren Sie die lokale Instanz bei CodeDeploy

Bei diesen Anweisungen gehen wir davon aus, dass Sie die Registrierung des lokalen Instance von der lokalen Instance aus durchführen. Sie können eine lokale Instanz von einem separaten Gerät oder einer separaten Instanz aus registrieren, auf der die Instanz AWS CLI installiert und konfiguriert ist, wie unter beschrieben. Schritt 5: Installieren und konfigurieren Sie den AWS CLI

Verwenden Sie den AWS CLI , um die lokale Instanz bei zu registrieren, CodeDeploy damit sie in Bereitstellungen verwendet werden kann.

1. Bevor Sie den verwenden können AWS CLI, benötigen Sie den Benutzer-ARN des IAM-Benutzers, in dem Sie ihn erstellt haben. Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance Wenn Sie noch nicht über den Benutzer-ARN verfügen, rufen Sie den Befehl get-user auf, geben Sie den Namen des IAM-Benutzers (mit der `--user-name` Option) an und fragen Sie nur den Benutzer-ARN ab (mit den `--query` Optionen und): `--output`

   ```
   aws iam get-user --user-name CodeDeployUser-OnPrem --query "User.Arn" --output text
   ```

2. Rufen Sie den Befehl https://docs.aws.amazon.com/cli/latest/reference/deploy/register-on-premises-instance.html register-on-premises-instance auf und geben Sie dabei Folgendes an:

   - Ein eindeutiger Bezeichner für die lokale Instance (mit der Option `--instance-name`).

     > **Important**
     >
     > Um die lokale Instance identifizieren zu können, insbesondere für Debugging-Zwecke, empfehlen wir dringend, dass Sie einen Namen angeben, die einige eindeutige Merkmale der lokalen Instance enthält (z. B. die Seriennummer oder eine eindeutige interne Asset-ID, falls zutreffend). Wenn Sie eine MAC-Adresse als Namen angeben, beachten Sie, dass MAC-Adressen Zeichen enthalten, die CodeDeploy nicht zulässig sind, z. B. Doppelpunkt (). `:` Eine Liste der zulässigen Zeichen finden Sie unter CodeDeploy Kontingente.

   - Die Benutzer-ARN des IAM-Benutzers, in dem Sie erstellt haben Schritt 1: Erstellen Sie einen IAM-Benutzer für die lokale Instance (mit der `--iam-user-arn` Option).

     Beispiel:

     ```
     aws deploy register-on-premises-instance --instance-name AssetTag12010298EX --
     iam-user-arn arn:aws:iam::444455556666:user/CodeDeployUser-OnPrem
     ```

Schritt 9: Taggen Sie die lokale Instanz

Sie können entweder die AWS CLI oder die CodeDeploy Konsole verwenden, um die lokale Instanz zu taggen. (CodeDeployverwendet lokale Instanz-Tags, um die Bereitstellungsziele während einer Bereitstellung zu identifizieren.)

So markieren Sie eine lokale Instance (CLI):

- Rufen Sie den Befehl https://docs.aws.amazon.com/cli/latest/reference/deploy/add-tags-to-on-premises-instances.html add-tags-to-on-premises-instances auf und geben Sie Folgendes an:

  - Der eindeutige Bezeichner für die lokale Instance (mit der Option `--instance-names`).
  - Der Name des Tag-Schlüssels und des Tag-Werts der lokalen Instance, die Sie verwenden möchten (mit der Option `--tags`). Sie müssen sowohl einen Namen als auch einen Wert angeben. CodeDeploy erlaubt keine lokalen Instanz-Tags, die nur Werte enthalten.

    Beispiel:

    ```
    aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
     --tags Key=Name,Value=CodeDeployDemo-OnPrem
    ```

So markieren Sie eine lokale Instance (Konsole):

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Wählen Sie im CodeDeploy Menü On-premises Instanzen aus.
3. Klicken Sie in der Liste der lokalen Instances auf den Pfeil neben der lokalen Instance, die Sie markieren möchten.
4. Wählen Sie in der Liste der Tags den gewünschten Tag-Schlüssel und -Wert aus oder geben Sie ihn ein. Nachdem Sie den Tag-Schlüssel und den Tag-Wert eingegeben haben, wird eine weitere Zeile angezeigt. Sie können das für bis zu 10 Tags wiederholen. Zum Entfernen eines Tags wählen Sie das Symbol "Löschen" aus ( ).
5. Wählen Sie nach dem Hinzufügen der Tags die Option Update Tags.

Schritt 10: Stellen Sie Anwendungsrevisionen auf der lokalen Instanz bereit

Sie können nun auf den registrierten und mit einem Tag markierten lokalen Instances Anwendungsrevisionen bereitstellen.

Sie stellen Anwendungsrevisionen auf lokalen Instances auf eine Weise bereit, die der Bereitstellung von Anwendungsrevisionen auf Amazon EC2-Instances ähnelt. Detaillierte Anweisungen finden Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy. Diese Anweisungen enthalten einen Link zu den Voraussetzungen, wie dem Erstellen einer Anwendung, Erstellen einer Bereitstellungsgruppe und Vorbereiten einer Anwendungsrevision. Wenn Sie eine einfache Beispielanwendungsrevision bereitstellen möchten, können Sie die unter Schritt 2: Erstellen Sie eine Version der Beispielanwendung im Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat Enterprise Linux) beschriebene Revision verwenden.

> **Important**
>
> Wenn Sie eine CodeDeploy Servicerolle wiederverwenden, um eine Bereitstellungsgruppe zu erstellen, die auf lokale Instances abzielt, müssen Sie dem `Action` Teil der `Tag:get*` Richtlinienerklärung der Servicerolle beifügen. Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.

Schritt 11: Verfolgen Sie die Bereitstellungen auf der lokalen Instanz

Nachdem Sie eine Anwendungsrevision auf einer registrierten und markierten lokalen Instance bereitgestellt haben, können Sie den Fortschritt der Bereitstellung nachverfolgen.

Sie verfolgen Bereitstellungen auf lokalen Instances auf eine Weise, die der Verfolgung von Bereitstellungen auf Amazon EC2-Instances ähnelt. Detaillierte Anweisungen finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen.

Verwaltung von Vorgängen für lokale Instanzen in CodeDeploy

Folgen Sie den Anweisungen in diesem Abschnitt, um Vorgänge auf Ihren lokalen Instances zu verwalten, nachdem Sie sie registriert haben CodeDeploy, z. B. um weitere Informationen zu lokalen Instances zu erhalten, Tags zu entfernen und sie zu deinstallieren und deren Registrierung aufzuheben.

Themen

- Informieren Sie sich über eine einzelne lokale Instanz
- Rufen Sie Informationen zu mehreren lokalen Instanzen ab
- Entfernen Sie manuell lokale Instanz-Tags aus einer lokalen Instanz
- Deinstallieren Sie den CodeDeploy Agenten automatisch und entfernen Sie die Konfigurationsdatei aus einer lokalen Instanz
- Automatische Deregistrierung einer lokalen Instanz
- Manuelles Aufheben der Registrierung einer lokalen Instanz

Informieren Sie sich über eine einzelne lokale Instanz

Sie können Informationen zu einer einzelnen lokalen Instance erhalten, indem Sie die Anweisungen unter CodeDeploy Bereitstellungsdetails anzeigen befolgen. Sie können die Konsole AWS CLI oder die CodeDeploy Konsole verwenden, um weitere Informationen zu einer einzelnen lokalen Instanz zu erhalten.

Abrufen von Informationen über eine einzelne lokale Instance (CLI)

- Rufen Sie den Befehl get-on-premises-instance auf und geben Sie den Namen an, der die lokale Instanz eindeutig identifiziert (mit der Option): `--instance-name`

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

Abrufen von Informationen über eine einzelne lokale Instance (Konsole)

1. Melden Sie sich bei der an und öffnen Sie die Konsole unter AWS-Managementkonsole . CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie On-premises Instanzen aus.
3. Wählen Sie in der Liste der lokalen Instances den Namen einer lokalen Instance aus, um ihre Details anzuzeigen.

Rufen Sie Informationen zu mehreren lokalen Instanzen ab

Sie können Informationen zu lokalen Instances erhalten, indem Sie die Anweisungen unter CodeDeploy Bereitstellungsdetails anzeigen befolgen. Sie können die Konsole AWS CLI oder die CodeDeploy Konsole verwenden, um weitere Informationen zu lokalen Instanzen zu erhalten.

Abrufen von Informationen über mehrere lokale Instances (CLI)

1. Rufen Sie den Befehl list-on-premises-instances auf, um eine Liste der lokalen Instanznamen zu erhalten. Geben Sie dabei Folgendes an:

   - Ob Informationen zu allen registrierten lokalen Instances oder allen lokalen Instances, deren Registrierung aufgehoben wurde, abgerufen werden sollen (mit der Option `--registration-status` und `Registered` bzw. `Deregistered`). Wenn Sie diese Angabe auslassen, werden Namen der registrierten lokalen Instances und der lokalen Instances, deren Registrierung aufgehoben wurde, zurückgegeben.
   - Ob nur Informationen zu lokalen Instances mit spezifischen Tags für lokale Instances abgerufen werden sollen (mit der Option `--tag-filters`). Geben Sie für jedes Tag für lokale Instances `Key`, `Value` und `Type` an (immer `KEY_AND_VALUE`). Trennen Sie mehrere Tags für lokale Instances durch Leerzeichen zwischen jeder Dreiergruppe aus `Key`, `Value` und `Type`.

   Beispiel:

   ```
   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. Für detailliertere Informationen rufen Sie den Befehl batch-get-on-premises-instances mit den Namen der lokalen Instanzen auf (mit der Option): `--instance-names`

   ```
   aws deploy batch-get-on-premises-instances --instance-names AssetTag12010298EX
    AssetTag09920444EX
   ```

Abrufen von Informationen über mehrere lokale Instances (Konsole)

1. Melden Sie sich bei der an und öffnen Sie die Konsole unter. AWS-Managementkonsole CodeDeploy https://console.aws.amazon.com/codedeploy

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie On-premises Instanzen aus.

   Informationen zu den lokalen Instances werden angezeigt.

Entfernen Sie manuell lokale Instanz-Tags aus einer lokalen Instanz

In der Regel entfernen Sie ein Tag für lokale Instances von einer lokalen Instance, wenn dieses nicht mehr genutzt wird oder wenn Sie die lokale Instance aus Bereitstellungsgruppen entfernen möchten, die dieses Tag benötigen. Sie können die AWS CLI oder die AWS CodeDeploy Konsole verwenden, um lokale Instanz-Tags aus lokalen Instanzen zu entfernen.

Sie müssen Tags für lokale Instances nicht von einer lokalen Instance entfernen, bevor Sie deren Registrierung aufheben.

Durch manuelles Entfernen von Tags für lokale Instances von einer lokalen Instance wird die Registrierung der Instance nicht aufgehoben. Der CodeDeploy Agent wird dadurch nicht von der Instanz deinstalliert. Die Konfigurationsdatei wird nicht von der Instance entfernt. Der mit der Instanz verknüpfte IAM-Benutzer wird nicht gelöscht.

Informationen zum automatischen Aufheben der Registrierung der lokalen Instance finden Sie unter Automatische Deregistrierung einer lokalen Instanz.

Informationen zum manuellen Aufheben der Registrierung der lokalen Instance finden Sie unter Manuelles Aufheben der Registrierung einer lokalen Instanz.

Informationen zur automatischen Deinstallation des CodeDeploy Agenten und zum Entfernen der Konfigurationsdatei aus der lokalen Instanz finden Sie unter. Deinstallieren Sie den CodeDeploy Agenten automatisch und entfernen Sie die Konfigurationsdatei aus einer lokalen Instanz

Informationen zur manuellen Deinstallation nur des CodeDeploy Agenten von der lokalen Instanz finden Sie unter. Verwaltung des CodeDeploy Agentenbetriebs

Informationen zum manuellen Löschen des zugehörigen IAM-Benutzers finden Sie unter Löschen eines IAM-Benutzers aus Ihrem Konto. AWS

Entfernen von Tags für lokale Instances von einer lokalen Instance (CLI)

- Rufen Sie die remove-tags-from-on-premises-instances auf und geben Sie dabei Folgendes an:

  - Die eindeutigen Namen der lokalen Instance (mit der Option `--instance-names`).
  - Die Namen und Werte der Tags, die Sie entfernen möchten (mit der Option `--tags`).

  Beispiel:

```
aws deploy remove-tags-from-on-premises-instances --instance-names
 AssetTag12010298EX --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Entfernen von Tags für lokale Instances von einer lokalen Instance (Konsole)

1. Melden Sie sich bei der an und öffnen Sie die Konsole unter. AWS-Managementkonsole CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie On-premises Instanzen aus.
3. Klicken Sie in der Liste der lokalen Instances auf den Namen der lokalen Instance, aus der Sie Tags entfernen möchten.
4. Wählen Sie im Bereich Tags neben jedem Tag, das Sie entfernen möchten, Remove (Entfernen) aus.
5. Wählen Sie nach dem Löschen der Tags die Option Update tags.

Deinstallieren Sie den CodeDeploy Agenten automatisch und entfernen Sie die Konfigurationsdatei aus einer lokalen Instanz

In der Regel deinstallieren Sie den CodeDeploy Agenten und entfernen die Konfigurationsdatei aus einer lokalen Instanz, nachdem Sie die Bereitstellung auf dieser Instanz nicht mehr planen.

> **Note**
>
> Durch die automatische Deinstallation des CodeDeploy Agenten und das Entfernen der Konfigurationsdatei aus einer lokalen Instanz wird die Registrierung einer lokalen Instanz nicht aufgehoben. Die Zugehörigkeit von Tags für lokale Instances, die mit der lokalen Instance verbunden sind, wird nicht aufgehoben. Der mit der lokalen Instanz verknüpfte IAM-Benutzer wird dadurch nicht gelöscht.
> Informationen zum automatischen Aufheben der Registrierung der lokalen Instance finden Sie unter Automatische Deregistrierung einer lokalen Instanz.
> Informationen zum manuellen Aufheben der Registrierung der lokalen Instance finden Sie unter Manuelles Aufheben der Registrierung einer lokalen Instanz.
> Informationen zum manuellen Aufheben der Zugehörigkeit zugehöriger Tags für lokale Instances finden Sie unter Entfernen Sie manuell lokale Instanz-Tags aus einer lokalen Instanz.
> Informationen zur manuellen Deinstallation des CodeDeploy Agenten von der lokalen Instanz finden Sie unter. Verwaltung des CodeDeploy Agentenbetriebs
> Informationen zum manuellen Löschen des zugehörigen IAM-Benutzers finden Sie unter Löschen eines IAM-Benutzers aus Ihrem Konto. AWS

Rufen Sie in der lokalen Instanz den Befehl AWS CLI uninstall auf.

Beispiel:

```
aws deploy uninstall
```

Der Befehl uninstall hat folgende Auswirkungen:

1. Stoppt den laufenden CodeDeploy Agenten auf der lokalen Instanz.
2. Deinstalliert den CodeDeploy Agenten von der lokalen Instanz aus.
3. Entfernt die Konfigurationsdatei von der lokalen Instance. (Für Ubuntu Server und RHEL ist dies/. `/etc/codedeploy-agent/conf` `codedeploy.onpremises.yml` Für Windows Server ist dies`C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.)

Automatische Deregistrierung einer lokalen Instanz

In der Regel heben Sie die Registrierung einer lokalen Instance auf, wenn Sie sie voraussichtlich nicht mehr bereitstellen werden. Wenn Sie die Registrierung einer lokalen Instance aufheben, wird diese, auch wenn sie Teil der Tags für lokale Instances einer Bereitstellungsgruppe ist, nicht mehr in Bereitstellungen eingeschlossen. Sie können das verwenden, AWS CLI um lokale Instanzen abzumelden.

> **Note**
>
> Sie können die CodeDeploy Konsole nicht verwenden, um die Registrierung einer lokalen Instanz aufzuheben. Zudem werden durch die Aufhebung der Registrierung einer lokalen Instance alle lokalen Instance-Tags, die mit der lokalen Instance verbunden sind, entfernt. Der CodeDeploy Agent wird nicht von der lokalen Instanz deinstalliert. Die Konfigurationsdatei der lokalen Instance wird nicht von der lokalen Instance entfernt.
> Informationen dazu, wie Sie mit der CodeDeploy Konsole einige (aber nicht alle) der Aktivitäten in diesem Abschnitt ausführen können, finden Sie im CodeDeploy Konsolenabschnitt vonManuelles Aufheben der Registrierung einer lokalen Instanz.
> Informationen zum manuellen Aufheben der Zugehörigkeit zugehöriger Tags für lokale Instances finden Sie unter Entfernen Sie manuell lokale Instanz-Tags aus einer lokalen Instanz.
> Informationen zur automatischen Deinstallation des CodeDeploy Agenten und zum Entfernen der Konfigurationsdatei aus der lokalen Instanz finden Sie unterDeinstallieren Sie den CodeDeploy Agenten automatisch und entfernen Sie die Konfigurationsdatei aus einer lokalen Instanz.
> Informationen zur manuellen Deinstallation nur des CodeDeploy Agenten von der lokalen Instanz finden Sie unter. Verwaltung des CodeDeploy Agentenbetriebs

Rufen Sie AWS CLI mit dem den Befehl deregister auf und geben Sie Folgendes an:

- Der Name, der die lokale Instanz eindeutig identifiziert CodeDeploy (mit der `--instance-name` Option).
- Optional, ob der IAM-Benutzer gelöscht werden soll, der der lokalen Instanz zugeordnet ist. Das Standardverhalten besteht darin, den IAM-Benutzer zu löschen. Wenn Sie den mit der lokalen Instance verbundenen IAM-Benutzer nicht löschen möchten, geben Sie im Befehl die Option `--no-delete-iam-user` an.

- Optional die AWS Region, in der die lokale Instanz registriert wurde CodeDeploy (mit der `--region` Option). Dies muss eine der unterstützten Regionen sein, die unter Region aufgeführt sind, und Endpunkte in Allgemeine AWS-Referenz(z. B.`us-west-2`). Wenn diese Option nicht angegeben ist, wird die AWS Standardregion verwendet, die dem aufrufenden IAM-Benutzer zugeordnet ist.

Ein Beispiel, das die Registrierung einer Instance aufhebt und den Benutzer löscht:

```
aws deploy deregister --instance-name AssetTag12010298EX --region us-west-2
```

Ein Beispiel, das die Registrierung einer Instance aufhebt und den Benutzer nicht löscht:

```
aws deploy deregister --instance-name AssetTag12010298EX --no-delete-iam-user --region
 us-west-2
```

Der Befehl deregister hat folgende Auswirkungen:

1. Deregistriert die lokale Instanz bei. CodeDeploy
2. Falls angegeben, wird der IAM-Benutzer gelöscht, der der lokalen Instanz zugeordnet ist.

Nach dem Aufheben der Registrierung einer lokalen Instance:

- Sie wird sofort nicht mehr auf der Konsole angezeigt.
- Sie können umgehend eine andere Instance mit demselben Namen erstellen.

Wenn dieser Befehl Fehler erkennt, wird eine Fehlermeldung angezeigt, in der beschrieben wird, wie Sie die verbleibenden Schritte manuell durchführen können. Andernfalls wird eine Meldung über den erfolgreichen Abschluss angezeigt, in der beschrieben wird, wie Sie den Befehl uninstall aufrufen.

Manuelles Aufheben der Registrierung einer lokalen Instanz

In der Regel heben Sie die Registrierung einer lokalen Instance auf, wenn Sie sie voraussichtlich nicht mehr bereitstellen werden. Sie verwenden die, AWS CLI um lokale Instanzen manuell abzumelden.

Durch die manuelle Deregistrierung einer lokalen Instanz wird der Agent nicht deinstalliert. CodeDeploy Die Konfigurationsdatei wird nicht von der Instance entfernt. Der mit der Instanz

verknüpfte IAM-Benutzer wird dadurch nicht gelöscht. Mit der Instance verbundene Tags werden nicht entfernt.

Informationen zur automatischen Deinstallation des CodeDeploy Agenten und zum Entfernen der Konfigurationsdatei aus der lokalen Instanz finden Sie unter. Deinstallieren Sie den CodeDeploy Agenten automatisch und entfernen Sie die Konfigurationsdatei aus einer lokalen Instanz

Informationen zur manuellen Deinstallation nur des CodeDeploy Agenten finden Sie unterVerwaltung des CodeDeploy Agentenbetriebs.

Informationen zum manuellen Löschen des zugehörigen IAM-Benutzers finden Sie unter Löschen eines IAM-Benutzers aus Ihrem AWS Konto.

Informationen zum manuellen Entfernen nur der zugehörigen Tags für lokale Instances finden Sie unter Entfernen Sie manuell lokale Instanz-Tags aus einer lokalen Instanz.

- Rufen Sie den Befehl deregister-on-premises-instance auf und geben Sie den Namen an, der die lokale Instanz eindeutig identifiziert (mit der Option): `--instance-name`

  ```
  aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
  ```

  Nach dem Aufheben der Registrierung einer lokalen Instance:

  - Sie wird sofort nicht mehr auf der Konsole angezeigt.
  - Sie können umgehend eine andere Instance mit demselben Namen erstellen.

Instanzdetails anzeigen mit CodeDeploy

Sie können die CodeDeploy Konsole, die oder die CodeDeploy APIs verwenden AWS CLI, um Details zu den in einer Bereitstellung verwendeten Instances anzuzeigen.

Informationen zur Verwendung von CodeDeploy API-Aktionen zum Anzeigen von Instanzen finden Sie unter GetDeploymentInstanceListDeploymentInstances, und ListOnPremisesInstances.

Themen

- Instanzdetails anzeigen (Konsole)
- Instanzdetails anzeigen (CLI)

Instanzdetails anzeigen (Konsole)

So zeigen Sie Instance-Details an:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Bereitstellungen aus.

   > **Note**
   >
   > Wenn keine Einträge angezeigt werden, stellen Sie sicher, dass die richtige Region ausgewählt ist. Wählen Sie in der Navigationsleiste in der Regionsauswahl eine der Regionen aus, die unter Region und Endpoints in der aufgeführt sind. Allgemeine AWS-Referenz CodeDeploy wird nur in diesen Regionen unterstützt.

3. Um Bereitstellungsdetails anzuzeigen, wählen Sie die Bereitstellungs-ID für die Instance aus.
4. Auf der Seite der Bereitstellung können Sie alle Instances im Bereich Instance activity (Instance-Aktivität) anzeigen.
5. Um auf der Seite zu den Details der Bereitstellung Informationen zu einzelnen Bereitstellungslebenszyklusereignissen anzuzeigen, wählen Sie in der Spalte Events die Option View events aus.

   > **Note**
   >
   > Wenn bei einem Lebenszyklusereignis Failed angezeigt wird, wählen Sie auf der Seite zu den Details der Instance View logs, View in EC2 oder beides aus. Tipps zur Fehlerbehebung finden Sie unter Beheben von Instance-Problemen.

6. Wenn Sie weitere Informationen zu einer Amazon EC2 EC2-Instance sehen möchten, wählen Sie die ID der Instance in der Spalte Instance-ID aus.

Instanzdetails anzeigen (CLI)

Rufen Sie entweder AWS CLI den `get-deployment-instance` Befehl oder den Befehl auf, um Instanzdetails anzuzeigen. `list-deployment-instances`

Um Details zu einer einzelnen Instanz anzuzeigen, rufen Sie den Befehl get-deployment-instance auf und geben Sie Folgendes an:

- Die eindeutige Bereitstellungs-ID. Rufen Sie den Befehl list-deployments auf, um die Bereitstellungs-ID abzurufen.
- Die eindeutige Instance-ID. Rufen Sie den Befehl list-deployment-instances auf, um die Instanz-ID abzurufen.

Um eine Liste von IDs für in einer Bereitstellung verwendete Instanzen anzuzeigen, rufen Sie den Befehl list-deployment-instances auf und geben Sie Folgendes an:

- Die eindeutige Bereitstellungs-ID. Rufen Sie den Befehl list-deployments auf, um die Bereitstellungs-ID abzurufen.
- Optional, ob nur bestimmte Instance-IDs nach dem jeweiligen Bereitstellungsstatus eingeschlossen werden sollen. (Wenn nicht angegeben, werden alle entsprechenden Instance-IDs aufgelistet, unabhängig von deren Bereitstellungsstatus.)

CodeDeploy Zustand der Instanz

CodeDeploy überwacht den Integritätsstatus der Instances in einer Bereitstellungsgruppe. Bereitstellungen schlagen fehl, wenn die Anzahl fehlerfreier Instances unter die Mindestanzahl fehlerfreier Instances fällt, die für die Bereitstellungsgruppe während einer Bereitstellung angegeben wurden. Wenn beispielsweise 85 % der Instances während einer Bereitstellung fehlerfrei bleiben müssen und die Bereitstellungsgruppe 10 Instances enthält, schlägt die gesamte Bereitstellung fehl, wenn die Bereitstellung auf einer einzigen Instance fehlschlägt. Das liegt daran, dass die Anzahl der verfügbaren fehlerfreien Instanzen bereits auf 90% sinkt, wenn eine Instanz offline geschaltet wird, sodass die neueste Anwendungsversion installiert werden kann. Eine fehlerhafte Instance plus eine weitere offline geschaltete Instance bedeuten, dass nur 80 % der Instances fehlerfrei und verfügbar sind. CodeDeploy lässt die gesamte Bereitstellung fehlschlagen.

Denken Sie unbedingt daran, dass für den Erfolg einer allgemeinen Bereitstellung Folgendes gelten muss:

- CodeDeploy kann für jede Instanz in der Bereitstellung bereitgestellt werden.
- Es muss die Bereitstellung auf mindestens einer Instance erfolgreich sein. Das bedeutet, dass selbst wenn der minimale Wert für fehlerfreie Hosts 0 ist, die Bereitstellung auf mindestens einer Instanz erfolgreich sein muss (d. h. mindestens eine Instanz muss fehlerfrei sein), damit die gesamte Bereitstellung erfolgreich ist.

Themen

- Gesundheitsstatus
- Ungefähr die Mindestanzahl fehlerfreier Instanzen
- Ungefähr die Mindestanzahl fehlerfreier Instanzen pro Availability Zone

Gesundheitsstatus

CodeDeploy weist jeder Instanz zwei Integritätsstatuswerte zu: Versionsstatus und Instanzstatus.

Revisionszustand

Der Revisionszustand basiert auf der Anwendungsrevision, die derzeit auf der Instance installiert ist. Sie hat die folgenden Statuswerte:

- Aktuell: Die auf der Instance installierte Revision stimmt mit der Revision für die letzte erfolgreiche Bereitstellung der Bereitstellungsgruppe überein.
- Alt: Die auf der Instance installierte Revision stimmt mit einer älteren Version der Anwendung überein.
- Unbekannt: Die Anwendungsrevision wurde nicht erfolgreich auf der Instance installiert.

Zustand der Instanz

Der Instance-Zustand basiert darauf, ob Bereitstellungen auf einer Instance erfolgreich waren. Er hat die folgenden Werte:

- Fehlerfrei: Die letzte Bereitstellung auf der Instance war erfolgreich.
- Fehlerhaft: Der Versuch, eine Revision auf der Instance bereitzustellen, ist fehlgeschlagen, oder eine Revision wurde noch nicht auf der Instance bereitgestellt.

CodeDeploy verwendet Revisionsstatus und Instanzintegrität, um die Bereitstellung für die Instanzen der Bereitstellungsgruppe in der folgenden Reihenfolge zu planen:

1. Fehlerhafter Instance-Zustand.
2. Unbekannter Revisionszustand.
3. Alter Revisionszustand.
4. Aktueller Revisionszustand.

Wenn die gesamte Bereitstellung erfolgreich ist, wird die Revision aktualisiert und die Werte für den Zustand der Bereitstellungsgruppe werden entsprechend der neusten Bereitstellung aktualisiert.

- Alle aktuellen Instances mit erfolgreicher Bereitstellung behalten den Status „Aktuell“. Andernfalls wechselt ihr Status zu „Unbekannt“.
- Alle alten oder unbekannten Instances mit erfolgreicher Bereitstellung erhalten den Status „Aktuell“. Andernfalls behalten Sie den Status „Alt“ bzw. „Unbekannt“.
- Alle fehlerfreien Instances mit erfolgreicher Bereitstellung behalten den Status „Fehlerfrei“. Andernfalls wechselt ihr Status zu „Fehlerhaft“.
- Alle fehlerhaften Instances mit erfolgreicher Bereitstellung erhalten den Status „Fehlerfrei“. Andernfalls behalten Sie den Status „Fehlerhaft“.

Wenn die gesamte Bereitstellung fehlschlägt oder angehalten wird:

- Für jede Instanz, für die CodeDeploy versucht wurde, die Anwendungsrevision bereitzustellen, ist der Instanzstatus auf fehlerfrei oder fehlerhaft gesetzt, je nachdem, ob der Bereitstellungsversuch für diese Instanz erfolgreich war oder fehlgeschlagen ist.
- Für jede Instanz, für die CodeDeploy nicht versucht wurde, die Anwendungsrevision bereitzustellen, wird der aktuelle Instanzintegritätswert beibehalten.
- Die Revision der Bereitstellungsgruppe bleibt gleich.

Ungefähr die Mindestanzahl fehlerfreier Instanzen

Die erforderliche Mindestanzahl fehlerfreier Instances wird im Rahmen der Bereitstellungskonfiguration definiert.

> **Important**
>
> Während einer blue/green Bereitstellung gelten die Bereitstellungskonfiguration und der Mindestwert für fehlerfreie Hosts für Instanzen in der Ersatzumgebung, nicht für Instanzen

in der ursprünglichen Umgebung. Wenn jedoch die Registrierung von Instances in der Originalumgebung beim Load Balancer aufgehoben wird, wird die gesamte Bereitstellung als fehlgeschlagen gekennzeichnet, wenn die Registrierung nur einer einzigen Original-Instance nicht erfolgreich aufgehoben werden kann.

CodeDeploy stellt drei standardmäßige Bereitstellungskonfigurationen bereit, für die üblicherweise die Mindestwerte für fehlerfreie Hosts verwendet wurden:

Name der Standardbereitstellungskonfiguration	Vordefinierter Mindestwert für fehlerfreie Hosts
CodeDeployDefault.OneAtATime	1
CodeDeployDefault.HalfAtATime	50 %
CodeDeployDefault.AllAtOnce	0

Weitere Informationen zu Standardbereitstellungskonfigurationen finden Sie unter Arbeiten mit Bereitstellungskonfigurationen in CodeDeploy.

Sie können benutzerdefinierte Bereitstellungskonfigurationen erstellen CodeDeploy , um Ihre eigenen Mindestwerte für einen fehlerfreien Host zu definieren. Sie können diese Werte als Ganzzahlen oder Prozentwerte definieren, wenn Sie die folgenden Operationen verwenden:

- Zum Beispiel`minimum-healthy-hosts`, wenn Sie den Befehl create-deployment-config in der verwenden. AWS CLI
- Wie beim MinimumHealthyHostsDatentyp `Value` in der API. CodeDeploy
- Wie `MinimumHealthyHosts` wenn Sie AWS::CodeDeploy:: DeploymentConfig in einer CloudFormation Vorlage verwenden.

CodeDeploy ermöglicht es Ihnen, eine Mindestanzahl fehlerfreier Instanzen für die Bereitstellung für zwei Hauptzwecke anzugeben:

- Um bestimmen zu können, ob die gesamte Bereitstellung erfolgreich ist oder fehlschlägt. Die Bereitstellung ist erfolgreich, wenn die Anwendungsrevision erfolgreich auf mindestens der Mindestanzahl fehlerfreier Instances bereitgestellt wurde.

- Um die Anzahl an Instances bestimmen zu können, die während einer Bereitstellung fehlerfrei sein müssen, damit die Bereitstellung fortgesetzt wird.

Sie können die Mindestanzahl fehlerfreier Instances für Ihre Bereitstellungsgruppe als eine Anzahl von Instances oder als einen Prozentsatz der Gesamtzahl von Instances angeben. Wenn Sie einen Prozentsatz angeben, CodeDeploy rechnet der Prozentsatz zu Beginn der Bereitstellung in die entsprechende Anzahl von Instanzen um und rundet alle Bruchteile der Instanzen auf.

CodeDeploy verfolgt den Integritätsstatus der Instanzen der Bereitstellungsgruppe während des Bereitstellungsprozesses und bestimmt anhand der für die Bereitstellung angegebenen Mindestanzahl fehlerfreier Instanzen, ob die Bereitstellung fortgesetzt werden soll. Das Grundprinzip besagt, dass eine Bereitstellung nicht dazu führen darf, dass die Anzahl fehlerfreier Instances unter die angegebene Mindestanzahl fällt. Die einzige Ausnahme zu dieser Regel gilt, wenn eine Bereitstellungsgruppe zu Anfang weniger als die angegebene Mindestanzahl fehlerfreier Instances aufweist. In diesem Fall wird die Anzahl fehlerfreier Instances durch den Bereitstellungsvorgang nicht weiter reduziert.

> **Note**
>
> CodeDeploy versucht, die Bereitstellung für alle Instances in einer Bereitstellungsgruppe durchzuführen, auch für diejenigen, die sich derzeit im Status Gestoppt befinden. Bei der Berechnung der Mindestanzahl fehlerfreier Hosts hat eine angehaltene Instance dieselbe Wirkung wie eine fehlerhafte Instance. Um Bereitstellungsfehler aufgrund von zu vielen angehaltenen Instances zu beheben, starten Sie entweder Instances neu oder ändern Sie ihre Tags, um sie aus der Bereitstellungsgruppe auszuschließen.

CodeDeploy startet den Bereitstellungsprozess, indem versucht wird, die Anwendungsrevision für die fehlerhaften Instanzen der Bereitstellungsgruppe bereitzustellen. CodeDeploy Ändert bei jeder erfolgreichen Bereitstellung den Integritätsstatus der Instanz auf Fehlerfrei und fügt sie den fehlerfreien Instanzen der Bereitstellungsgruppe hinzu. CodeDeploy vergleicht dann die aktuelle Anzahl fehlerfreier Instances mit der angegebenen Mindestanzahl fehlerfreier Instances.

- Wenn die Anzahl der fehlerfreien Instanzen kleiner oder gleich der angegebenen Mindestanzahl fehlerfreier Instanzen ist, wird die Bereitstellung CodeDeploy abgebrochen, um sicherzustellen, dass die Anzahl der fehlerfreien Instanzen nicht mit mehr Bereitstellungen abnimmt.

- Wenn die Anzahl der fehlerfreien Instanzen die angegebene Mindestanzahl fehlerfreier Instanzen um mindestens eins übersteigt, wird die CodeDeploy Anwendungsrevision für die ursprüngliche Gruppe fehlerfreier Instanzen bereitgestellt.

Wenn eine Bereitstellung auf einer fehlerfreien Instance fehlschlägt, wird der Integritätsstatus dieser Instance in „Fehlerhaft“ CodeDeploy geändert. Im Verlauf der Bereitstellung wird die aktuelle Anzahl fehlerfreier Instanzen CodeDeploy aktualisiert und mit der angegebenen Mindestanzahl fehlerfreier Instanzen verglichen. Wenn die Anzahl der fehlerfreien Instanzen zu einem beliebigen Zeitpunkt des Bereitstellungsprozesses auf die angegebene Mindestanzahl fällt, wird die Bereitstellung CodeDeploy beendet. Dadurch wird verhindert, dass die nächste Bereitstellung fehlschlägt, weil die Anzahl fehlerfreier Instances unter die angegebene Mindestanzahl fällt.

> **Note**
>
> Achten Sie darauf, dass die angegebene Mindestanzahl fehlerfreier Instances kleiner ist als die Gesamtzahl der Instances in der Bereitstellungsgruppe. Wenn Sie einen Prozentsatz angeben, bedenken Sie, dass dieser aufgerundet wird. Andernfalls ist die Anzahl fehlerfreier Instances zu Beginn der Bereitstellung bereits kleiner oder gleich der angegebenen Mindestanzahl fehlerfreier Instances und CodeDeploy lässt die gesamte Bereitstellung umgehend fehlschlagen.

CodeDeploy verwendet außerdem die angegebene Mindestanzahl fehlerfreier Instanzen und die tatsächliche Anzahl fehlerfreier Instanzen, um zu bestimmen, ob und wie die Anwendungsversion für mehrere Instanzen bereitgestellt werden soll. Standardmäßig wird die Anwendungsrevision auf so vielen Instanzen wie möglich CodeDeploy bereitgestellt, ohne dass das Risiko besteht, dass die Anzahl der fehlerfreien Instanzen unter die angegebene Mindestanzahl fehlerfreier Instanzen fällt.

Um die Anzahl der Instanzen zu ermitteln, auf denen gleichzeitig bereitgestellt werden soll, CodeDeploy wird die folgende Berechnung verwendet:

```
[total-hosts] - [minimum-healthy-hosts] =
        [number-of-hosts-to-deploy-to-at-once]
```

Beispiel:

- Wenn Ihre Bereitstellungsgruppe aus 10 Instances besteht und Sie die Mindestanzahl fehlerfreier Instanzen auf 9 festlegen CodeDeploy , wird jeweils nur eine Instanz bereitgestellt.

- Wenn Ihre Bereitstellungsgruppe aus 10 Instances besteht und Sie die Mindestanzahl fehlerfreier Instances auf 3 festlegen, erfolgt die CodeDeploy Bereitstellung im ersten Batch auf 7 Instances gleichzeitig und im zweiten Batch auf die restlichen 3 Instances.
- Wenn Ihre Bereitstellungsgruppe aus 10 Instances besteht und Sie die Mindestanzahl fehlerfreier Instances auf 0 setzen, CodeDeploy erfolgt die Bereitstellung auf 10 Instances gleichzeitig.

Beispiele

In den folgenden Beispielen wird von einer Bereitstellungsgruppe mit 10 Instances ausgegangen.

Mindestanzahl fehlerfreier Instances: 95 %

CodeDeploy rundet die Mindestanzahl fehlerfreier Instanzen auf 10 Instanzen auf, was der Anzahl fehlerfreier Instanzen entspricht. Die gesamte Bereitstellung schlägt umgehend fehl, ohne dass die Revision auf einer einzigen Instance bereitgestellt wurde.

Mindestanzahl fehlerfreier Instances: 9

CodeDeploy stellt die Revision jeweils für eine Instanz bereit. Wenn die Bereitstellung auf einer der Instanzen fehlschlägt, schlägt die gesamte Bereitstellung CodeDeploy sofort fehl, da die Anzahl der fehlerfreien Instanzen der Mindestanzahl intakter Instanzen entspricht. Die Ausnahme zu dieser Regel besagt, dass wenn die letzte Instance fehlschlägt, die Bereitstellung trotzdem erfolgreich ist.

CodeDeploy setzt die Bereitstellung fort, eine Instanz nach der anderen, bis eine Bereitstellung fehlschlägt oder die gesamte Bereitstellung abgeschlossen ist. Wenn alle 10 Bereitstellungen erfolgreich sind, hat die Bereitstellungsgruppe jetzt 10 fehlerfreie Instances.

Mindestanzahl fehlerfreier Instances: 8

CodeDeploy stellt die Revision auf zwei Instanzen gleichzeitig bereit. Wenn zwei dieser Bereitstellungen fehlschlagen, schlägt die gesamte Bereitstellung CodeDeploy sofort fehl. Die Ausnahme zu dieser Regel besagt, dass wenn die letzte Instance die zweite ist, die fehlschlägt, die Bereitstellung trotzdem erfolgreich ist.

Mindestanzahl fehlerfreier Instances: 0

CodeDeploy stellt die Revision gleichzeitig für die gesamte Bereitstellungsgruppe bereit. Mindestens eine Bereitstellung einer Instance muss erfolgreich sein, damit die allgemeine Bereitstellung erfolgreich ist. Wenn 0 Instances fehlerfrei sind, schlägt die Bereitstellung fehl. Dies liegt an der Anforderung, dass mindestens eine Instanz nach Abschluss der gesamten

Bereitstellung fehlerfrei sein muss, um eine Gesamtbereitstellung als erfolgreich zu kennzeichnen, auch wenn der Mindestwert für fehlerfreie Instanzen 0 ist.

Ungefähr die Mindestanzahl fehlerfreier Instanzen pro Availability Zone

> **Note**
>
> In diesem Abschnitt werden die Begriffe Instance und Host synonym verwendet, um sich auf Amazon EC2 EC2-Instances zu beziehen.

Wenn Sie für Instances in mehreren Availability Zones bereitstellen, können Sie optional die zonal configuration Funktion aktivieren, mit der Sie die Bereitstellung jeweils in einer Availability Zone durchführen können CodeDeploy .

Wenn diese Funktion aktiviert ist, CodeDeploy wird sichergestellt, dass die Anzahl der fehlerfreien Hosts über den Werten „Mindestanzahl fehlerfreier Hosts pro Zone“ und „Mindestanzahl fehlerfreier Hosts“ liegt. Wenn die Anzahl der fehlerfreien Hosts unter einen der Werte fällt, schlägt die Bereitstellung in allen Availability CodeDeploy Zones fehl.

Um die Anzahl der Hosts zu berechnen, auf denen die Bereitstellung gleichzeitig erfolgen soll, werden sowohl die Werte „Mindestanzahl fehlerfreier Hosts pro Zone“ als auch die Werte für „mindestens fehlerfreie Hosts“ CodeDeploy verwendet. CodeDeploy verwendet die geringere Anzahl von Berechnungen `[A]` und, wo und `[B]` sind: `[A]` `[B]`

```
[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]
```

Nachdem die Anzahl der Hosts bestimmt wurde, auf denen die Bereitstellung gleichzeitig erfolgen soll CodeDeploy , erfolgt die Bereitstellung auf Hosts in Batches dieser Anzahl, jeweils eine Availability Zone, mit einer optionalen Pause (oder „Bake-Time“) zwischen den Zonen.

Beispiel

Wenn Ihre Bereitstellung wie folgt konfiguriert ist:

- `[total-hosts]` ist `200`
- `[minimum-healthy-hosts]` ist `160`
- `[total-hosts-per-AZ]` ist `100`
- `[minimum-healthy-hosts-per-AZ]` ist `50`

Dann …

- `[A]` = `200 - 160 = 40`
- `[B]` = `100 - 50 = 50`
- `40`ist weniger als `50`

CodeDeploy Wird daher sofort auf `40` Hosts bereitgestellt.

In diesem Szenario verläuft die Bereitstellung wie folgt:

1. CodeDeploy wird in der ersten Availability Zone bereitgestellt:
   a. CodeDeploy wird auf den ersten `40` Hosts bereitgestellt.
   b. CodeDeploy wird auf den nächsten `40` Hosts bereitgestellt.
   c. CodeDeploy wird auf den verbleibenden `20` Hosts bereitgestellt.

   Die Bereitstellung in der ersten Availability Zone ist jetzt abgeschlossen.
2. (Optional) CodeDeploy wartet, bis die Bereitstellung in der ersten Zone wie in der Einstellung Überwachungsdauer oder Monitordauer hinzufügen für die erste Zone festgelegt ist, „läuft“. Wenn es keine Probleme gibt, CodeDeploy fährt fort.
3. CodeDeploy wird in der zweiten Availability Zone bereitgestellt:
   a. CodeDeploy wird auf den ersten `40` Hosts bereitgestellt.
   b. CodeDeploy wird auf den nächsten `40` Hosts bereitgestellt.
   c. CodeDeploy wird auf den verbleibenden `20` Hosts bereitgestellt.

   Die Bereitstellung in der zweiten und letzten Availability Zone ist jetzt abgeschlossen.

Weitere Informationen zur Funktion für die zonale Konfiguration und zur Angabe der Mindestanzahl fehlerfreier Instances pro Availability Zone finden Sie unterzonal configuration.

Arbeiten mit Bereitstellungskonfigurationen in CodeDeploy

Eine Bereitstellungskonfiguration ist ein Satz von Regeln sowie Erfolgs- und Fehlerbedingungen. Sie wird von CodeDeploy während Bereitstellungen verwendet. Diese Regeln und Bedingungen sind unterschiedlich, je nachdem, ob Sie die Bereitstellung auf einer EC2/On-Premises Rechenplattform, einer AWS Lambda-Rechenplattform oder einer Amazon ECS-Rechenplattform durchführen.

Bereitstellungskonfigurationen auf einer EC2/on-premises Rechenplattform

Wenn Sie die Bereitstellung auf einer EC2/On-Premises Rechenplattform durchführen, gibt die Bereitstellungskonfiguration mithilfe eines Werts für mindestens fehlerfreie Hosts und eines optionalen Werts für fehlerfreie Hosts pro Zone die Anzahl oder den Prozentsatz der Instances an, die während einer Bereitstellung jederzeit verfügbar sein müssen.

Sie können eine der drei vordefinierten Bereitstellungskonfigurationen verwenden, die von bereitgestellt werden, AWS oder eine benutzerdefinierte Bereitstellungskonfiguration erstellen. Weitere Informationen zum Erstellen benutzerdefinierter Bereitstellungskonfigurationen finden Sie unterCreate a Deployment Configuration. Wenn Sie keine Bereitstellungskonfiguration angeben, CodeDeploy verwendet die CodeDeployDefault.OneAtATime Bereitstellungskonfiguration.

Weitere Informationen darüber, wie der Zustand einer Instanz während einer Bereitstellung CodeDeploy überwacht und bewertet wird, finden Sie unterInstance Health. Eine Liste der Bereitstellungskonfigurationen, die bereits für Ihr AWS Konto registriert sind, finden Sie unterView Deployment Configuration Details.

Vordefinierte Bereitstellungskonfigurationen für eine EC2/on-premises Rechenplattform

In der folgenden Tabelle sind die vordefinierten Bereitstellungskonfigurationen aufgeführt.

> **Note**
>
> Es gibt keine vordefinierten Bereitstellungskonfigurationen, die diese zonal configuration Funktion unterstützen (mit der Sie die Anzahl fehlerfreier Hosts pro Availability Zone

angeben können). Wenn Sie diese Funktion verwenden möchten, müssen Sie Ihre eigene Bereitstellungskonfiguration erstellen.

Bereitstellungskonfiguration	Description
CodeDeployDefault.AllAtOnce	In-place Bereitstellungen: Versucht, eine Anwendungsrevision in möglichst vielen Instances gleichzeitig bereitzustellen. Als Status der Gesamtbereitstellung wird Erfolgreich angezeigt, wenn die Anwendungsrevision in einer oder mehreren Instances bereitgestellt werden konnte. Es wird Fehlgeschlagen als Status der Gesamtbereitstellung angezeigt, wenn die Anwendungsrevision auf keiner Instance bereitgestellt werden konnte. Am Beispiel von neun Instanzen wird CodeDeployDefault.AllAtOnce versucht, die Bereitstellung auf allen neun Instanzen gleichzeitig durchzuführen. Die Gesamtbereitstellung ist erfolgreich, wenn die Revision in mindestens einer Instance bereitgestellt werden kann. Sie schlägt nur fehl, wenn die Revision in keiner der neun Instances bereitgestellt werden kann. Blue/green Bereitstellungen: • Bereitstellung in einer Ersatzumgebung: Es gelten dieselben Bereitstellungsregeln wie CodeDeployDefault.AllAtOnce für Bereitstellungen vor Ort. • Datenverkehrsumleitung: Leitet den Datenverkehr in alle Instances in der Austauschumgebung gleichzeitig um. Ist erfolgreich, wenn der Datenverkehr in mindestens eine Instance umgeleitet werden

Bereitstellungskonfiguration	Description
	kann. Schlägt fehl, wenn der Datenverkehr in keine der Instances umgeleitet werden kann.

Bereitstellungskonfiguration	Description
CodeDeployDefault.HalfAtATime	In-place Bereitstellungen: Stellt die Anwendungsrevision maximal in der Hälfte der Instances gleichzeitig bereit (bei Nachkommastellen wird abgerundet). Die Gesamtbereitstellung ist erfolgreich, wenn die Anwendungsrevision in mindestens der Hälfte der Instances bereitgestellt werden kann (bei Nachkommastellen wird aufgerund et). Andernfalls schlägt die Bereitstellung fehl. Beispiel: Bei neun Instances wird die Revision in maximal vier Instances gleichzeitig bereitges tellt. Die Gesamtbereitstellung ist erfolgreich, wenn die Revision in mindestens fünf Instances bereitgestellt werden kann. Andernfalls schlägt die Bereitstellung fehl. **Note** Wenn Sie für Instances in mehreren Auto Scaling Scaling-Gruppen bereitste llen, CodeDeploy erfolgt die Bereitste llung auf bis zu der Hälfte der Instances gleichzeitig, unabhängig von der Auto Scaling Scaling-Gruppe, in der sie sich befinden. Nehmen wir zum Beispiel an, Sie haben zwei Auto Scaling Scaling-G ruppen mit jeweils 10 Instances. `ASG1` `ASG2` In diesem Szenario CodeDeplo y könnten Sie in nur 10 Instances bereitstellen `ASG1` und dies als Erfolg betrachten, da es auf mindestens der Hälfte der Instances bereitgestellt wurde.

Bereitstellungskonfiguration	Description
	Blue/green Bereitstellungen: • Bereitstellung in einer Ersatzumgebung: Es gelten dieselben Bereitstellungsregeln wie CodeDeployDefault.HalfAtATime für Bereitstellungen vor Ort. • Datenverkehrsumleitung: Leitet den Datenverkehr in maximal die Hälfte der in der Austauschumgebung vorhandenen Instances gleichzeitig um. Ist erfolgreich, wenn der Datenverkehr in mindestens die Hälfte der Instances umgeleitet werden kann. Andernfal ls schlägt fehl.

Bereitstellungskonfiguration	Description
CodeDeployDefault.OneAtATime	In-place Bereitstellungen: Stellt die Anwendungsrevision nacheinander auf jeweils nur einer Instance bereit. Für Bereitstellungsgruppen mit mehr als einer Instance gilt: • Die Gesamtbereitstellung ist erfolgreich, wenn die Anwendungsrevision in allen Instances bereitgestellt werden kann. Ausnahme: Wenn die Revision in der letzten Instance nicht bereitgestellt werden kann, gilt die Gesamtbereitstellung dennoch als erfolgreich. Dies liegt daran CodeDeploy , dass mit der CodeDeployDefault.OneAtATime Konfiguration jeweils nur eine Instanz offline geschaltet werden kann. • Die Gesamtbereitstellung schlägt fehl, sobald die Anwendungsrevision in einer anderen als der letzten Instance nicht bereitgestellt werden kann. • Beispiel: Bei neun Instances wird die Revision nacheinander jeweils in einer Instance bereitgestellt. Die Gesamtbereitstellung ist erfolgreich, wenn die Revision in den ersten acht Instances bereitgestellt werden kann. Die Gesamtbereitstellung schlägt fehl, wenn die Revision in einer der ersten acht Instances nicht bereitgestellt werden kann. In Bereitstellungsgruppen mit einer einzigen Instance ist die Gesamtbereitstellung nur

Bereitstellungskonfiguration	Description
	erfolgreich, wenn die Revision auf dieser einen Instance bereitgestellt werden kann. Blue/green Bereitstellungen: • Bereitstellung in einer Ersatzumgebung: Es gelten dieselben Bereitstellungsregeln wie CodeDeployDefault.OneAtATime für Bereitstellungen vor Ort. • Datenverkehrsumleitung: Leitet den Datenverkehr nacheinander in jeweils eine der Instances in der Austauschumgebung um. Ist erfolgreich, wenn der Datenverkehr in alle Austausch-Instances umgeleitet werden kann. Schlägt fehl, sobald die Umleitung erstmals fehlschlägt. Ausnahme: Wenn die letzte Instance nicht registriert werden kann, gilt die Gesamtbereitstellung dennoch als erfolgreich.

Bereitstellungskonfigurationen auf einer Amazon-ECS-Datenverarbeitungsplattform

Wenn Sie die Bereitstellung auf einer Amazon ECS-Rechenplattform durchführen, gibt die Bereitstellungskonfiguration an, wie der Datenverkehr auf den aktualisierten Amazon ECS-Aufgabensatz übertragen wird. Sie können den Datenverkehr mithilfe einer kanarischen, linearen oder All-in-Once-Bereitstellungskonfiguration verlagern. Weitere Informationen finden Sie unter Bereitstellungskonfiguration.

Sie können auch eine eigene benutzerdefinierte Canary- oder lineare Bereitstellungskonfiguration erstellen. Weitere Informationen finden Sie unter Create a Deployment Configuration.

Vordefinierte Bereitstellungskonfigurationen für eine Amazon ECS-Rechenplattform

In der folgenden Tabelle sind die vordefinierten Konfigurationen aufgeführt, die für Amazon ECS-Bereitstellungen verfügbar sind.

> **Note**
>
> Wenn Sie einen Network Load Balancer verwenden, wird nur die `CodeDeployDefault.ECSAllAtOnce` vordefinierte Bereitstellungskonfiguration unterstützt.

Bereitstellungskonfiguration	Description
CodeDeployDefault.ECSLinear10PercentEvery1Minutes	Verschiebt jede Minute 10 Prozent des Datenverkehrs, bis der gesamte Datenverkehr verschoben ist.
CodeDeployDefault.ECSLinear10PercentEvery3Minutes	Verschiebt alle drei Minuten 10 Prozent des Datenverkehrs, bis der gesamte Datenverkehr verschoben ist.
CodeDeployDefault.ECSCanary10Percent5Minutes	Verschiebt 10 Prozent des Datenverkehrs im ersten Inkrementschritt. Die restlichen 90 Prozent werden fünf Minuten später bereitgestellt.
CodeDeployDefault.ECSCanary10Percent15Minutes	Verschiebt 10 Prozent des Datenverkehrs im ersten Inkrementschritt. Die restlichen 90 Prozent werden 15 Minuten später bereitgestellt.
CodeDeployDefault.ECSAllAtOnce	Verschiebt den gesamten Datenverkehr auf einmal in den aktualisierten Amazon ECS-Container.

Bereitstellungskonfigurationen für CloudFormation blue/green Bereitstellungen (Amazon ECS)

Wenn Sie die Bereitstellung über Bereitstellungen auf einer Amazon ECS-Rechenplattform durchführen, gibt die CloudFormation blue/green Bereitstellungskonfiguration an, wie der Datenverkehr auf den aktualisierten Amazon ECS-Container verlagert wird. Sie können den Datenverkehr mithilfe einer kanarischen, linearen oder All-in-Once-Bereitstellungskonfiguration verlagern. Weitere Informationen finden Sie unter Bereitstellungskonfiguration.

Bei CloudFormation blue/green Bereitstellungen können Sie keine eigene benutzerdefinierte kanarische oder lineare Bereitstellungskonfiguration erstellen. Eine schrittweise Anleitung CloudFormation zur Verwaltung Ihrer Amazon blue/green ECS-Bereitstellungen finden Sie unter Automatisieren von blue/green ECS-Bereitstellungen CodeDeploy mithilfe CloudFormation im CloudFormation Benutzerhandbuch.

> **Note**
>
> Die Verwaltung von Amazon blue/green ECS-Bereitstellungen mit CloudFormation ist in den Regionen Europa (Mailand), Afrika (Kapstadt) und Asien-Pazifik (Osaka) nicht verfügbar.

Bereitstellungskonfigurationen auf einem AWS Lambda Plattform für die Datenverarbeitung

Wenn Sie auf einer AWS Lambda-Rechenplattform bereitstellen, gibt die Bereitstellungskonfiguration an, wie der Datenverkehr auf die neuen Lambda-Funktionsversionen in Ihrer Anwendung verlagert wird. Sie können den Datenverkehr mithilfe einer kanarischen, linearen oder All-in-Once-Bereitstellungskonfiguration verlagern. Weitere Informationen finden Sie unter Bereitstellungskonfiguration.

Sie können auch eine eigene benutzerdefinierte Canary- oder lineare Bereitstellungskonfiguration erstellen. Weitere Informationen finden Sie unter Create a Deployment Configuration.

Vordefinierte Bereitstellungskonfigurationen für ein AWS Lambda Plattform für die Datenverarbeitung

In der folgenden Tabelle sind die vordefinierten Konfigurationen für AWS Lambda -Bereitstellungen aufgelistet.

Bereitstellungskonfiguration	Description
CodeDeployDefault.LambdaCanary10Percent5Minutes	Verschiebt 10 Prozent des Datenverkehrs im ersten Inkrementschritt. Die restlichen 90 Prozent werden fünf Minuten später bereitgestellt.
CodeDeployDefault.LambdaCanary10Percent10Minutes	Verschiebt 10 Prozent des Datenverkehrs im ersten Inkrementschritt. Die restlichen 90 Prozent werden 10 Minuten später bereitgestellt.
CodeDeployDefault.LambdaCanary10Percent15Minutes	Verschiebt 10 Prozent des Datenverkehrs im ersten Inkrementschritt. Die restlichen 90 Prozent werden 15 Minuten später bereitgestellt.
CodeDeployDefault.LambdaCanary10Percent30Minutes	Verschiebt 10 Prozent des Datenverkehrs im ersten Inkrementschritt. Die restlichen 90 Prozent werden 30 Minuten später bereitgestellt.
CodeDeployDefault.LambdaLinear10PercentEvery1Minute	Verschiebt jede Minute 10 Prozent des Datenverkehrs, bis der gesamte Datenverkehr verschoben ist.
CodeDeployDefault.LambdaLinear10PercentEvery2Minutes	Verschiebt alle zwei Minuten 10 Prozent des Datenverkehrs, bis der gesamte Datenverkehr verschoben ist.

Bereitstellungskonfiguration	Description
CodeDeployDefault.LambdaLinear10PercentEvery3Minutes	Verschiebt alle drei Minuten 10 Prozent des Datenverkehrs, bis der gesamte Datenverkehr verschoben ist.
CodeDeployDefault.LambdaLinear10PercentEvery10Minutes	Verschiebt alle 10 Minuten 10 Prozent des Datenverkehrs, bis der gesamte Datenverkehr verschoben ist.
CodeDeployDefault.LambdaAllAtOnce	Leitet den gesamten Datenverkehr auf einmal auf die aktualisierten Lambda-Funktionen um.

Topics

- Create a Deployment Configuration
- View Deployment Configuration Details
- Delete a Deployment Configuration

Erstellen Sie eine Bereitstellungskonfiguration mit CodeDeploy

Wenn Sie keine der mitgelieferten Standardbereitstellungskonfigurationen verwenden möchten CodeDeploy, können Sie mithilfe der folgenden Anweisungen Ihre eigene erstellen.

Sie können die CodeDeploy Konsole AWS CLI, die CodeDeploy APIs oder eine CloudFormation Vorlage verwenden, um benutzerdefinierte Bereitstellungskonfigurationen zu erstellen.

Informationen zur Verwendung einer CloudFormation Vorlage zum Erstellen einer Bereitstellungskonfiguration finden Sie unterCloudFormation Vorlagen als CodeDeploy Referenz.

Themen

- Eine Bereitstellungskonfiguration erstellen (Konsole)
- Erstellen einer Bereitstellungskonfiguration mit CodeDeploy (AWS CLI)

Eine Bereitstellungskonfiguration erstellen (Konsole)

Verwenden Sie die folgenden Anweisungen, um eine Bereitstellungskonfiguration mithilfe der AWS Konsole zu erstellen.

So erstellen Sie eine Bereitstellungskonfiguration CodeDeploy mithilfe der Konsole

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Wählen Sie im Navigationsbereich Bereitstellungskonfigurationen aus.

   Eine Liste der integrierten Bereitstellungskonfigurationen wird angezeigt.
3. Klicken Sie auf Create deployment configuration (Bereitstellungskonfiguration erstellen).
4. Geben Sie im Feld Name der Bereitstellungskonfiguration einen Namen für die Bereitstellungskonfiguration ein. Beispiel, **`my-deployment-config`**.
5. Wählen Sie unter Rechenplattform eine der folgenden Optionen aus:

   - EC2/On-premises
   - AWS Lambda
   - Amazon ECS
6. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Sie EC2/On-premises gewählt haben:
     1. Geben Sie unter Mindestanzahl fehlerfreier Hosts die Anzahl oder den Prozentsatz der Instanzen an, die während einer Bereitstellung jederzeit verfügbar bleiben müssen. Weitere Informationen dazu, wie CodeDeploy während einer Bereitstellung den Instance-Zustand überwacht und evaluiert, erhalten Sie unter Instance Health.
     2. (Optional) Wählen Sie unter Zonale Konfiguration die Option Zonale Konfiguration aktivieren aus, um Ihre Anwendung jeweils in einer Availability Zone innerhalb einer AWS Region CodeDeploy bereitzustellen. Durch die Bereitstellung in jeweils einer Availability Zone können Sie Ihre Bereitstellung einem immer größeren Publikum zugänglich machen, wenn

das Vertrauen in die Leistung und Durchführbarkeit der Bereitstellung zunimmt. Wenn Sie keine zonale Konfiguration aktivieren, wird Ihre Anwendung auf einer zufälligen Auswahl von Hosts in einer Region CodeDeploy bereitgestellt.

Wenn Sie die Funktion für die zonale Konfiguration aktivieren, beachten Sie Folgendes:

- Die Funktion zur zonalen Konfiguration wird nur bei In-Place-Bereitstellungen auf Amazon EC2 EC2-Instances unterstützt. (Blue/greenBereitstellungen und lokale Instances werden nicht unterstützt.) Weitere Informationen zu In-Situ-Bereitstellungen finden Sie unter Bereitstellungstyp.
- Die Funktion zur zonalen Konfiguration wird bei vordefinierten Bereitstellungskonfigurationen nicht unterstützt. Um eine zonale Konfiguration zu verwenden, müssen Sie eine benutzerdefinierte Bereitstellungskonfiguration erstellen, wie hier beschrieben.
- Wenn eine Bereitstellung rückgängig gemacht werden CodeDeploy muss, CodeDeploy werden die Rollback-Operationen auf zufälligen Hosts ausgeführt. (CodeDeployführt nicht erwartungsgemäß ein Rollback für eine Zone nach der anderen durch.) Dieses Rollback-Verhalten wurde aus Leistungsgründen gewählt. Weitere Informationen zu Rollbacks finden Sie unter. Eine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy

3. Wenn Sie das Kontrollkästchen Zonale Konfiguration aktivieren aktiviert haben, geben Sie optional die folgenden Optionen an:
   - (Optional) Geben Sie im Feld Monitordauer den Zeitraum in Sekunden an, der nach Abschluss einer Bereitstellung in einer Availability Zone warten CodeDeploy muss. CodeDeploy wartet diese Zeit, bevor eine Bereitstellung in der nächsten Availability Zone gestartet wird. Erwägen Sie, eine Monitordauer hinzuzufügen, um der Bereitstellung etwas Zeit zu geben, sich in einer Availability Zone zu bewähren (oder zu „backen“), bevor sie in der nächsten Zone veröffentlicht wird. Wenn Sie keine Monitordauer angeben, wird sofort mit der Bereitstellung in der nächsten Availability Zone CodeDeploy begonnen. Weitere Informationen zur Funktionsweise der Einstellung „Monitordauer“ finden Sie unterUngefähr die Mindestanzahl fehlerfreier Instanzen pro Availability Zone.
   - (Optional) Wählen Sie Monitordauer für die erste Zone hinzufügen aus, um eine Monitordauer festzulegen, die nur für die erste Availability Zone gilt. Sie können diese Option festlegen, wenn Sie zusätzliche Backzeit für die erste Availability Zone einplanen möchten. Wenn Sie unter Monitordauer für erste Zone hinzufügen keinen Wert angeben, wird der Wert für die Dauer der Überwachung für die erste Availability Zone CodeDeploy verwendet.

- (Optional) Geben Sie unter Mindestanzahl fehlerfreier Hosts pro Zone die Anzahl oder den Prozentsatz der Instanzen an, die während einer Bereitstellung pro Availability Zone verfügbar bleiben müssen. Wählen Sie FLEET_PERCENT, um einen Prozentsatz anzugeben, oder HOST_COUNT, um eine Zahl anzugeben. Dieses Feld funktioniert in Verbindung mit dem Feld Minimum gesunder Hosts. Weitere Informationen finden Sie unter Ungefähr die Mindestanzahl fehlerfreier Instanzen pro Availability Zone.

  Wenn Sie unter Mindestanzahl fehlerfreier Hosts pro Zone keinen Wert angeben, wird der Standardwert `0` Prozent CodeDeploy verwendet.

- Wenn Sie sich für Amazon ECS entschieden haben AWS Lambda:

  1. Wählen Sie als Typ Linear oder Canary aus.
  2. Führen Sie in den Feldern Schritt und Intervall einen der folgenden Schritte aus:
     - Wenn Sie Canary ausgewählt haben, geben Sie für Schritt einen Prozentsatz des Traffics zwischen 1 und 99 ein, der verschoben werden soll. Dies ist der Prozentsatz des Traffics, der in der ersten Stufe verschoben wird. Der verbleibende Datenverkehr wird nach dem gewählten Intervall im zweiten Inkrementschritt verschoben.

       Geben Sie für Intervall die Anzahl der Minuten zwischen der ersten und zweiten Verkehrsschicht ein.
     - Wenn Sie Linear wählen, geben Sie für Schritt einen Prozentsatz des Verkehrs zwischen 1 und 99 ein, der verschoben werden soll. Dies ist der Prozentsatz des Datenverkehrs, der zu Beginn jedes Intervalls verschoben wird.

       Geben Sie für Intervall die Anzahl der Minuten zwischen den einzelnen inkrementellen Schichten ein.

7. Klicken Sie auf Create deployment configuration (Bereitstellungskonfiguration erstellen).

   Sie haben jetzt eine Bereitstellungskonfiguration, die Sie einer Bereitstellungsgruppe zuordnen können.

Erstellen einer Bereitstellungskonfiguration mit CodeDeploy (AWS CLI)

Rufen Sie den Befehl create-deployment-config auf, AWS CLI um mit dem eine Bereitstellungskonfiguration zu erstellen.

Im folgenden Beispiel wird eine EC2/On-Premises Bereitstellungskonfiguration mit dem Namen erstelltThreeQuartersHealthy, dass 75% der Zielinstanzen während einer Bereitstellung fehlerfrei bleiben müssen:

```
aws deploy create-deployment-config --deployment-config-name ThreeQuartersHealthy --
minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

Im folgenden Beispiel wird eine EC2/On-Premises Bereitstellungskonfiguration mit dem Namen erstellt300Total50PerAZ, nach der insgesamt 300 Zielinstanzen pro Bereitstellung und 50 pro Availability Zone fehlerfrei bleiben müssen. Außerdem wird eine Monitordauer von 1 Stunde festgelegt.

```
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}}'
```

Im folgenden Beispiel wird eine AWS Lambda-Bereitstellungskonfiguration mit dem Namen Canary25Percent45Minutes erstellt. Sie verwendet die Canary-Datenverkehrsverschiebung, um 25 Prozent des Datenverkehrs im ersten Inkrementschritt zu verschieben. Die restlichen 75 Prozent werden 45 Minuten später verschoben:

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
 --traffic-routing-config
 "type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform Lambda
```

Im folgenden Beispiel wird eine Amazon ECS-Bereitstellungskonfiguration mit dem Namen erstelltCanary25Percent45Minutes. Sie verwendet die Canary-Datenverkehrsverschiebung, um 25 Prozent des Datenverkehrs im ersten Inkrementschritt zu verschieben. Die restlichen 75 Prozent werden 45 Minuten später verschoben:

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
 --traffic-routing-config
 "type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform ECS
```

Details zur Bereitstellungskonfiguration anzeigen mit CodeDeploy

Sie können die CodeDeploy Konsole, die oder die CodeDeploy APIs verwenden AWS CLI, um Details zu den mit Ihrem AWS Konto verknüpften Bereitstellungskonfigurationen anzuzeigen. Eine Beschreibung der vordefinierten CodeDeploy Bereitstellungskonfigurationen finden Sie unterVordefinierte Bereitstellungskonfigurationen für eine EC2/on-premises Rechenplattform.

Themen

- Details zur Bereitstellungskonfiguration anzeigen (Konsole)
- Bereitstellungskonfiguration anzeigen (CLI)

Details zur Bereitstellungskonfiguration anzeigen (Konsole)

So verwenden Sie die CodeDeploy Konsole, um eine Liste der Namen der Bereitstellungskonfiguration anzuzeigen:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie Bereitstellungskonfigurationen aus.

   Hier können Sie die Bereitstellungskonfigurationsnamen und die Kriterien der einzelnen Bereitstellungskonfigurationen anzeigen.

   > **Note**
   >
   > Wenn keine Einträge angezeigt werden, stellen Sie sicher, dass die richtige Region ausgewählt ist. Wählen Sie in der Navigationsleiste in der Regionsauswahl eine der Regionen aus, die unter Region und Endpoints in der aufgeführt sind. Allgemeine AWS-Referenz CodeDeploy wird nur in diesen Regionen unterstützt.

Bereitstellungskonfiguration anzeigen (CLI)

Rufen Sie entweder AWS CLI den `get-deployment-config` Befehl oder den Befehl auf, um Details zur Bereitstellungskonfiguration anzuzeigen. `list-deployment-configs`

Um Details zu einer einzelnen Bereitstellungskonfiguration anzuzeigen, rufen Sie den Befehl get-deployment-config auf und geben Sie dabei den eindeutigen Namen der Bereitstellungskonfiguration an.

Rufen Sie den Befehl list-deployments auf, um Details zu mehreren Bereitstellungskonfigurationen anzuzeigen.

Löschen Sie eine Bereitstellungskonfiguration mit CodeDeploy

Sie können die AWS CLI oder die CodeDeploy APIs verwenden, um benutzerdefinierte Bereitstellungskonfigurationen zu löschen, die mit Ihrem AWS Konto verknüpft sind. Sie können nicht integrierte Bereitstellungskonfigurationen löschen, z. B. `CodeDeployDefault.AllAtOnce`, `CodeDeployDefault.HalfAtATime` und `CodeDeployDefault.OneAtATime`.

> **Warning**
>
> Sie können keine benutzerdefinierten Bereitstellungskonfigurationen löschen, die noch verwendet werden. Wenn Sie eine ungenutzte, benutzerdefinierte Bereitstellungskonfiguration löschen, können Sie sie nicht mehr mit neuen Bereitstellungen und neuen Bereitstellungsgruppen verknüpfen. Diese Aktion kann nicht rückgängig gemacht werden.

Um eine Bereitstellungskonfiguration AWS CLI zu löschen, rufen Sie den Befehl delete-deployment-config auf und geben Sie dabei den Namen der Bereitstellungskonfiguration an. Rufen Sie den Befehl list-deployment-configs auf, um eine Liste der Namen der Bereitstellungskonfiguration anzuzeigen.

Im folgenden Beispiel wird eine Bereitstellungskonfiguration mit dem Namen gelöscht. ThreeQuartersHealthy

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Arbeiten mit Anwendungen in CodeDeploy

Nach der Konfiguration von Instances, aber bevor Sie eine Revision bereitstellen können, müssen Sie eine Anwendung in CodeDeploy erstellen. Eine Anwendung ist einfach ein Name oder Container, der von CodeDeploy verwendet wird, um sicherzustellen, dass während einer Bereitstellung die richtige Revision, Bereitstellungskonfiguration und Bereitstellungsgruppe referenziert werden.

Verwenden Sie die Informationen in der folgenden Tabelle für nächste Schritte:

Plattform für die Datenverarbeitung	Szenario	Informationen für den nächsten Schritt
EC2/On-Premises	Ich habe noch keine Instances erstellt.	Lesen Sie Arbeiten mit Instanzen für CodeDeploy und kehren Sie dann zu dieser Seite zurück.
EC2/On-Premises	Ich habe Instances erstellt, deren Markierung mit Tags aber noch nicht abgeschlossen.	Lesen Sie Tagging Instances for Deployments und kehren Sie dann zu dieser Seite zurück.
EC2/On-Premises, AWS Lambda und Amazon ECS	Ich habe noch keine Anwendung erstellt.	Siehe Erstellen Sie eine Anwendung mit CodeDeploy
EC2/On-Premises, AWS Lambda und Amazon ECS	Ich habe bereits eine Anwendung, aber keine Bereitstellungsgruppe erstellt.	Siehe Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy.
EC2/On-Premises, AWS Lambda und Amazon ECS	Ich habe bereits eine Anwendung und Bereitstellungsgruppe erstellt, aber noch keine Anwendungsrevision.	Siehe Arbeiten mit Anwendungsrevisionen für CodeDeploy.
EC2/On-Premises, AWS Lambda und Amazon ECS	Ich habe bereits eine Anwendung und Bereitstellungsgruppe erstellt und	Siehe Erstellen Sie eine Bereitstellung mit CodeDeploy.

Plattform für die Datenverarbeitung	Szenario	Informationen für den nächsten Schritt
	ich habe meine Anwendungsrevision bereits hochgeladen. Ich bin zur Bereitstellung bereit.	

Themen

- Erstellen Sie eine Anwendung mit CodeDeploy
- Anwendungsdetails anzeigen mit CodeDeploy
- Erstellen einer Benachrichtigungsregel
- Eine CodeDeploy Anwendung umbenennen
- Löschen Sie eine Anwendung in CodeDeploy

Erstellen Sie eine Anwendung mit CodeDeploy

Eine Anwendung ist einfach ein Name oder ein Container, der verwendet wird, CodeDeploy um sicherzustellen, dass während einer Bereitstellung auf die richtige Version, Bereitstellungskonfiguration und Bereitstellungsgruppe verwiesen wird. Sie können die CodeDeploy Konsole, die AWS CLI, die CodeDeploy APIs oder eine CloudFormation Vorlage verwenden, um Anwendungen zu erstellen.

Ihr Code oder Ihre Anwendungsversion wird mithilfe eines Prozesses, der als Bereitstellung bezeichnet wird, auf Instances installiert. CodeDeploy unterstützt zwei Arten von Bereitstellungen:

- In-place Bereitstellung: Die Anwendung auf jeder Instanz in der Bereitstellungsgruppe wird gestoppt, die neueste Anwendungsversion wird installiert und die neue Version der Anwendung wird gestartet und validiert. Sie können einen Load Balancer verwenden, sodass jede Instanz während der Bereitstellung deregistriert und nach Abschluss der Bereitstellung wieder betriebsbereit ist. Nur Bereitstellungen, die die EC2/On-Premises Rechenplattform verwenden, können direkte Bereitstellungen verwenden. Weitere Informationen zu In-Situ-Bereitstellungen finden Sie unter Überblick über eine direkte Bereitstellung.
- Blue/green Bereitstellung: Das Verhalten Ihrer Bereitstellung hängt davon ab, welche Rechenplattform Sie verwenden:

- Blue/green auf einer EC2/On-Premises Rechenplattform: Die Instanzen in einer Bereitstellungsgruppe (der ursprünglichen Umgebung) werden mithilfe der folgenden Schritte durch eine andere Gruppe von Instanzen (die Ersatzumgebung) ersetzt:
  - Instanzen werden für die Ersatzumgebung bereitgestellt.
  - Die neueste Anwendungsversion ist auf den Ersatzinstanzen installiert.
  - Für Aktivitäten wie Anwendungstests und Systemverifizierung fällt eine optionale Wartezeit an.
  - Instances in der Ersatzumgebung werden bei einem oder mehreren Elastic Load Balancing Balancing-Load Balancing-Load Balancern registriert, wodurch der Datenverkehr zu ihnen umgeleitet wird. Instances in der ursprünglichen Umgebung werden deregistriert und können beendet oder für andere Zwecke weiterlaufen.

  > **Note**
  >
  > Wenn Sie eine EC2/On-Premises Rechenplattform verwenden, beachten Sie, dass blue/green Bereitstellungen nur mit Amazon EC2 EC2-Instances funktionieren.

- Blue/green auf einer AWS Lambda oder Amazon ECS-Rechenplattform: Der Datenverkehr wird schrittweise entsprechend einer kanarischen, linearen oder All-in-Once-Bereitstellungskonfiguration verschoben.
- Blue/green Bereitstellungen durch CloudFormation: Der Datenverkehr wird im Rahmen eines Stack-Updates von Ihren aktuellen Ressourcen auf Ihre aktualisierten Ressourcen verlagert. CloudFormation Derzeit werden nur blue/green ECS-Bereitstellungen unterstützt.

Weitere Informationen zu blue/green Bereitstellungen finden Sie unter. Überblick über eine Bereitstellung blue/green

Wenn Sie die CodeDeploy Konsole verwenden, um eine Anwendung zu erstellen, konfigurieren Sie gleichzeitig ihre erste Bereitstellungsgruppe. Wenn Sie die verwenden, AWS CLI um eine Anwendung zu erstellen, erstellen Sie ihre erste Bereitstellungsgruppe in einem separaten Schritt.

Eine Liste der Anwendungen, die bereits für Ihr AWS Konto registriert sind, finden Sie unterAnwendungsdetails anzeigen mit CodeDeploy. Informationen zur Verwendung einer CloudFormation Vorlage zum Erstellen einer Anwendung finden Sie unterCloudFormation Vorlagen als CodeDeploy Referenz.

Beide Bereitstellungstypen sind nicht für alle Ziele geeignet. In der folgenden Tabelle ist aufgelistet, welche Bereitstellungstypen mit Bereitstellungen für die drei verschiedenen Bereitstellungsziele funktionieren.

Bereitstellungsziel	In-place	Blue/green
Amazon EC2	Ja	Ja
On-premises	Ja	Nein
Serverlose AWS Lambda-Funktionen	Nein	Ja
Amazon ECS-Anwendungen	Nein	Ja

Themen

- Erstellen Sie eine Anwendung für eine direkte Bereitstellung (Konsole)
- Erstellen Sie eine Anwendung für eine blue/green Bereitstellung (Konsole)
- Eine Anwendung für eine Amazon ECS-Servicebereitstellung erstellen (Konsole)
- Erstellen Sie eine Anwendung für ein AWS Lambda Bereitstellung von Funktionen (Konsole)
- Anwendung erstellen (CLI)

Erstellen Sie eine Anwendung für eine direkte Bereitstellung (Konsole)

So verwenden Sie die CodeDeploy Konsole, um eine Anwendung für eine direkte Bereitstellung zu erstellen:

> **Warning**
>
> Führen Sie diese Schritte nicht aus, wenn:
>
> - Sie haben Ihre Instances nicht für die Verwendung in CodeDeploy Bereitstellungen vorbereitet. Befolgen Sie die Anweisung unter Arbeiten mit Instanzen für CodeDeploy und befolgen Sie anschließend die Schritte dieses Themas, um Ihre Instances einzurichten.

- Sie möchten eine Anwendung mit einer benutzerdefinierten Bereitstellungskonfiguration erstellen, aber Sie haben die Bereitstellungskonfiguration noch nicht erstellt. Befolgen Sie die Anweisungen unter Create a Deployment Configuration und befolgen Sie anschließend die Schritte dieses Themas.
- Sie verfügen nicht über eine Servicerolle, die über das erforderliche Mindestmaß an Vertrauen und Berechtigungen verfügt. CodeDeploy Um eine Servicerolle mit den erforderlichen Berechtigungen zu erstellen und zu konfigurieren, befolgen Sie die Anweisungen unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy und kehren Sie dann zu den Schritten in diesem Thema zurück.
- Sie möchten einen Classic Load Balancer, Application Load Balancer oder Network Load Balancer in Elastic Load Balancing für die direkte Bereitstellung auswählen, haben ihn aber noch nicht erstellt.

So erstellen Sie mithilfe der Konsole eine Anwendung für eine direkte Bereitstellung: CodeDeploy

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich den Abschnitt Deploy (Bereitstellen) und wählen Sie Getting started (Erste Schritte) aus.
3. Wählen Sie Create application aus.
4. Geben Sie unter Application name (Anwendungsname) den Namen Ihrer Anwendung ein.
5. Wählen Sie unter Compute Platform die Option EC2/On-premises.
6. Wählen Sie Create application aus.
7. Wählen Sie auf Ihrer Anwendungsseite über die Registerkarte Deployment groups (Bereitstellungsgruppen) die Option Create deployment group (Bereitstellungsgruppe erstellen) aus.
8. Geben Sie im Feld Deployment group name (Name der Bereitstellungsgruppe) einen Namen ein, der die Bereitstellungsgruppe beschreibt.

> **Note**
>
> Wenn Sie dieselben Einstellungen verwenden möchten, die in einer anderen Bereitstellungsgruppe verwendet wurden (einschließlich des Namens der Bereitstellungsgruppe, der Tags, der Amazon EC2 Auto Scaling Scaling-Gruppennamen oder beides sowie der Bereitstellungskonfiguration), geben Sie diese Einstellungen auf dieser Seite an. Obwohl diese neue Bereitstellungsgruppe und die bestehende Bereitstellungsgruppe denselben Namen haben, werden sie als separate Bereitstellungsgruppen CodeDeploy behandelt, da sie jeweils separaten Anwendungen zugeordnet sind.

9. Wählen Sie im Feld Service role (Servicerolle) eine Servicerolle aus, die CodeDeploy-Zugriff auf Ihre Ziel-Instance gewährt.
10. Wählen Sie unter Bereitstellungstyp die Option aus In-place.
11. Wählen Sie unter Environment configuration (Umgebungskonfiguration) eine der folgenden Optionen aus:

    a. Amazon EC2 Auto Scaling Scaling-Gruppen: Geben Sie den Namen einer Amazon EC2 Auto Scaling Scaling-Gruppe ein, für die Sie Ihre Anwendungsrevision bereitstellen möchten, oder wählen Sie ihn aus. Wenn neue Amazon EC2-Instances als Teil einer Amazon EC2 Auto Scaling-Gruppe gestartet werden, CodeDeploy kann Ihre Revisionen automatisch für die neuen Instances bereitgestellt werden. Sie können einer Bereitstellungsgruppe bis zu 10 Amazon EC2 Auto Scaling Scaling-Gruppen hinzufügen.

    b. Amazon EC2 EC2-Instances oder On-premisesInstances: Geben Sie in die Felder Schlüssel und Wert die Werte des Schlüssel-Wert-Paares ein, mit dem Sie die Instances markiert haben. Sie können bis zu 10 Schlüssel-Wert-Paare in einer einzigen Tag-Gruppe markieren.

        i. Sie können Platzhalter im Feld Wert verwenden, um alle Instances zu identifizieren, die nach bestimmten Mustern gekennzeichnet sind, z. B. ähnliche Amazon EC2 EC2-Instance-, Kostenstellen- und Gruppennamen usw. Wenn Sie beispielsweise Name im Feld Schlüssel wählen und in das Feld Wert eingeben**`GRP-*a`**, werden alle Instances CodeDeploy identifiziert, die diesem Muster entsprechen, z. B. **`GRP-1aGRP-2a`**, und. **`GRP-XYZ-a`**

        ii. Das Feld Value (Wert) unterscheidet Groß- und Kleinschreibung.

        iii. Klicken Sie auf Remove tag (Tag entfernen), um ein Schlüssel-Wert-Paar aus der Liste zu entfernen.

Wenn CodeDeploy Instances gefunden werden, die jedem angegebenen Schlüssel-Wert-Paar oder Amazon EC2 Auto Scaling Scaling-Gruppennamen entsprechen, wird die Anzahl der passenden Instances angezeigt. Klicken Sie auf die Zahl, um weitere Informationen zu den Instances zu erhalten.

Wenn Sie die Kriterien für die bereitgestellten Instances verfeinern möchten, wählen Sie Tag-Gruppe hinzufügen, um eine Tag-Gruppe zu erstellen. Sie können bis zu drei Tag-Gruppen mit jeweils bis zu zehn Schlüssel-Wert-Paaren erstellen. Wenn Sie mehrere Tag-Gruppen in einer Bereitstellungsgruppe verwenden, gehören nur Instances, die von allen tag-Gruppen identifiziert werden, zu der Bereitstellungsgruppe. Dies bedeutet, dass eine Instance mindestens einem Tag in jeder der Gruppen entsprechen muss, um zu der Bereitstellungsgruppe zu gehören.

Informationen zur Verwendung von Tag-Gruppen zur Verfeinerung Ihrer Bereitstellungsgruppe finden Sie unter Tagging Instances for Deployments.

12. Wählen Sie unter Deployment settings (Bereitstellungseinstellungen) eine Bereitstellungskonfiguration aus, um die Rate zu steuern, mit der Ihre Anwendung für Instances bereitgestellt wird (z. B. jeweils nur eine oder mehrere gleichzeitig). Weitere Informationen zu Bereitstellungskonfigurationen finden Sie unter Arbeiten mit Bereitstellungskonfigurationen in CodeDeploy.
13. (Optional) Wählen Sie unter Load Balancer die Option Load Balancing aktivieren aus und wählen Sie dann aus den Listen die Classic Load Balancers, Application Load Balancer Balancer-Zielgruppen und Network Load Balancer Balancer-Zielgruppen aus, um den Datenverkehr zu den Instances während der Bereitstellung zu verwalten. CodeDeploy Sie können bis zu 10 Classic Load Balancer und 10 Zielgruppen auswählen, also insgesamt 20 Elemente. Stellen Sie sicher, dass die Amazon EC2 EC2-Instances, für die Sie die Bereitstellung durchführen möchten, bei den ausgewählten Load Balancers (Classic Load Balancers) oder Zielgruppen (Application Load Balancers und Network Load Balancers) registriert sind.

    Während einer Bereitstellung werden die ursprünglichen Instances von den ausgewählten Load Balancern und Zielgruppen abgemeldet, um zu verhindern, dass während der Bereitstellung Traffic an diese Instances weitergeleitet wird. Wenn die Bereitstellung abgeschlossen ist, wird jede Instanz bei allen ausgewählten Classic Load Balancern und Zielgruppen erneut registriert.

    Weitere Informationen zu Load Balancern für CodeDeploy Bereitstellungen finden Sie unter. Integrating CodeDeploy with Elastic Load Balancing

14. (Optional) Erweitern Sie Erweitert und konfigurieren Sie alle Optionen, die Sie in die Bereitstellung einbeziehen möchten, z. B. Amazon SNS SNS-Benachrichtigungsauslöser, CloudWatch Amazon-Alarme oder automatische Rollbacks.

    Weitere Informationen finden Sie unter Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe.
15. Wählen Si Create deployment group (Bereitstellungsgruppe erstellen).

Der nächste Schritt besteht in der Vorbereitung einer Revision zur Bereitstellung für die Anwendung und Bereitstellungsgruppe. Detaillierte Anweisungen finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.

Erstellen Sie eine Anwendung für eine blue/green Bereitstellung (Konsole)

So verwenden Sie die CodeDeploy Konsole, um eine Anwendung für eine blue/green Bereitstellung zu erstellen:

> **Note**
>
> Eine Bereitstellung auf der AWS Lambda-Rechenplattform ist immer eine blue/green Bereitstellung. Sie geben keine Option für den Bereitstellungstyp an.

> **Warning**
>
> Führen Sie diese Schritte nicht aus, wenn:
>
> - Sie haben keine Instanzen, auf denen der CodeDeploy Agent installiert ist und den Sie während des blue/green Bereitstellungsprozesses ersetzen möchten. Befolgen Sie die Anweisung unter Arbeiten mit Instanzen für CodeDeploy und befolgen Sie anschließend die Schritte dieses Themas, um Ihre Instances einzurichten.
> - Sie möchten eine Anwendung mit einer benutzerdefinierten Bereitstellungskonfiguration erstellen, aber Sie haben die Bereitstellungskonfiguration noch nicht erstellt. Befolgen Sie die Anweisungen unter Create a Deployment Configuration und befolgen Sie anschließend die Schritte dieses Themas.
> - Sie haben keine Servicerolle, die mindestens den unter beschriebenen Vertrauensstellungen und Berechtigungen vertraut CodeDeploy . Schritt 2: Erstellen Sie

eine Servicerolle für CodeDeploy Um eine Servicerolle zu erstellen und zu konfigurieren, befolgen Sie die Anweisungen unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy und befolgen Sie dann die Schritte dieses Themas.

- Sie haben in Elastic Load Balancing keinen Classic Load Balancer, Application Load Balancer oder Network Load Balancer für die Registrierung der Instances in Ihrer Ersatzumgebung erstellt. Weitere Informationen finden Sie unter Richten Sie in Elastic Load Balancing einen Load Balancer für CodeDeploy Amazon EC2 EC2-Bereitstellungen ein.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich den Abschnitt Deploy (Bereitstellen) und wählen Sie Getting started (Erste Schritte) aus.
3. Geben Sie unter Application name (Anwendungsname) den Namen Ihrer Anwendung ein.
4. Wählen Sie unter Compute platform (Plattform für die Datenverarbeitung) die Option EC2/On-Premises aus.
5. Wählen Sie Create application aus.
6. Wählen Sie auf Ihrer Anwendungsseite über die Registerkarte Deployment groups (Bereitstellungsgruppen) die Option Create deployment group (Bereitstellungsgruppe erstellen) aus.
7. Geben Sie im Feld Deployment group name (Name der Bereitstellungsgruppe) einen Namen ein, der die Bereitstellungsgruppe beschreibt.

   > **Note**
   >
   > Wenn Sie dieselben Einstellungen verwenden möchten, die in einer anderen Bereitstellungsgruppe verwendet wurden (einschließlich der Namens-Tags für die Bereitstellungsgruppe, Amazon EC2 Auto Scaling Scaling-Gruppennamen und der Bereitstellungskonfiguration), wählen Sie diese Einstellungen auf dieser

Seite aus. Obwohl diese neue Bereitstellungsgruppe und die vorhandene Bereitstellungsgruppe denselben Namen haben, behandelt CodeDeploy sie als separate Bereitstellungsgruppen, da beide mit einer separaten Anwendung verknüpft sind.

8. Wählen Sie im Feld Service role (Servicerolle) eine Servicerolle aus, die CodeDeploy-Zugriff auf Ihre Ziel-Instance gewährt.
9. Wählen Sie unter Bereitstellungstyp die Option Blue/green.
10. Wählen Sie unter Environment configuration, welche Methode Sie verwenden möchten, um die Instances für die Austauschumgebung bereitzustellen:

    a. Amazon EC2 Auto Scaling Scaling-Gruppe automatisch kopieren: CodeDeploy Erstellt eine Amazon EC2 Auto Scaling Scaling-Gruppe, indem eine von Ihnen angegebene kopiert wird.

    b. Manually provision instances: Sie müssen erst eine Bereitstellung erstellen, bevor Sie Instances für Ihre Austauschumgebung angeben können. Sie müssen die Instances erstellen, bevor Sie die Bereitstellung starten. Stattdessen geben Sie hier die Instances an, die Sie hier ersetzen möchten.

11. Abhängig von Ihrer Wahl in Schritt 10, führen Sie einen der folgenden Schritte aus:

    - Wenn Sie Amazon EC2 Auto Scaling Scaling-Gruppe automatisch kopieren ausgewählt haben: Wählen Sie in der Amazon EC2 Auto Scaling Scaling-Gruppe den Namen der Amazon EC2 Auto Scaling Scaling-Gruppe, die Sie als Vorlage für die Amazon EC2 Auto Scaling Scaling-Gruppe für die Instances in Ihrer Ersatzumgebung verwenden möchten, oder geben Sie ihn ein. Die Anzahl der aktuell fehlerfreien Instances in der von Ihnen ausgewählten Amazon EC2 Auto Scaling Scaling-Gruppe wird in Ihrer Ersatzumgebung erstellt.
    - Wenn Sie Instances manuell bereitstellen ausgewählt haben: Aktivieren Sie Amazon EC2 Auto Scaling Scaling-Gruppen, Amazon EC2-Instances oder beides, um Instances anzugeben, die zu dieser Bereitstellungsgruppe hinzugefügt werden sollen. Geben Sie Amazon EC2-Tag-Werte oder Amazon EC2 Auto Scaling Scaling-Gruppennamen ein, um die Instances in Ihrer ursprünglichen Umgebung zu identifizieren (d. h. die Instances, die Sie ersetzen möchten oder auf denen die aktuelle Anwendungsrevision ausgeführt wird).

12. Wählen Sie unter Load Balancer die Option Load Balancing aktivieren aus und wählen Sie dann aus den Listen die Classic Load Balancers, Application Load Balancer Balancer-Zielgruppen und Network Load Balancer Balancer-Zielgruppen aus, bei denen Sie Ihre Amazon EC2 EC2-Ersatz-Instances registrieren möchten. Jede Ersatz-Instance wird bei allen ausgewählten Classic Load Balancern und Zielgruppen registriert. Sie können bis zu 10 Classic Load Balancer und 10 Zielgruppen auswählen, also insgesamt 20 Artikel.

Der Datenverkehr wird entsprechend den von Ihnen ausgewählten Konfigurationseinstellungen für die Datenverkehrsumleitung und Bereitstellung von den ursprünglichen zu den Ersatz-Instances umgeleitet.

Weitere Informationen zu Load Balancern für CodeDeploy Bereitstellungen finden Sie unter. Integrating CodeDeploy with Elastic Load Balancing

13. Prüfen Sie unter Deployment settings die Standardoptionen für die Umleitung des Datenverkehrs an die Austauschumgebung, welche Bereitstellungskonfiguration Sie für die Bereitstellung verwenden möchten und wie Instances in der ursprünglichen Umgebung nach der Bereitstellung behandelt werden.

    Wenn Sie die Einstellungen ändern möchten, fahren Sie mit dem nächsten Schritt fort. Andernfalls fahren Sie mit Schritt 15 fort.

14. Um die Bereitstellungseinstellungen für die blue/green Bereitstellung zu ändern, ändern Sie eine der folgenden Einstellungen.

Einstellung	Optionen
Traffic rerouting	• Sofortige Umleitung des Datenverkehrs: Sobald Instances in der Ersatzumgebung bereitgestellt und die neueste Anwendungsversion auf ihnen installiert ist, werden sie automatisch bei den angegebenen Load Balancern und Zielgruppen registriert, sodass der Datenverkehr an diese umgeleitet wird. Instances in der ursprünglichen Umgebung werden dann abgemeldet. • Ich entscheide, ob der Datenverkehr umgeleitet werden soll: Instances in der Ersatzumgebung werden nicht bei den angegebenen Load Balancern und Zielgruppen registriert, es sei denn, Sie leiten den Datenverkehr manuell um. Wenn die von Ihnen angegebene Wartezeit abläuft, ohne dass Datenverk

Einstellung	Optionen
	ehr umgeleitet wurde, wird der Bereitste llungsstatus zu "Stopped" geändert.
Bereitstellungskonfiguration	Wählen Sie die Rate, mit der Instances in der Ersatzumgebung bei den Load Balancern und Zielgruppen registriert werden, z. B. einzeln oder alle gleichzeitig. **Note** Nachdem der Datenverkehr erfolgrei ch zu der Austauchumgebung umgeleitet wurde, werden die Instances in der ursprünglichen Umgebung unabhängig von der gewählten Bereitstellungskonfiguratio n alle gleichzeitig abgemeldet. Weitere Informationen finden Sie unter Arbeiten mit Bereitstellungskonfigurationen in CodeDeploy.

Einstellung	Optionen
Original instances	• Beenden Sie die ursprünglichen Instances in der Bereitstellungsgruppe: Nachdem der Datenverkehr in die Ersatzumgebung umgeleitet wurde, werden die Instances, die von den Load Balancern und Zielgrupp en abgemeldet wurden, nach der von Ihnen angegebenen Wartezeit beendet. • Lassen Sie die ursprünglichen Instanzen in der Bereitstellungsgruppe laufen: Nachdem der Datenverkehr in die Ersatzumgebung umgeleitet wurde, werden die Instances, die von den Load Balancern und Zielgruppen abgemeldet wurden, weiter ausgeführt.

15. (Optional) Konfigurieren Sie unter Erweitert Optionen, die Sie in die Bereitstellung einbeziehen möchten, z. B. Amazon SNS SNS-Benachrichtigungsauslöser, CloudWatch Amazon-Alarme oder automatische Rollbacks.

    Weitere Informationen zum Festlegen erweiterter Optionen für Bereitstellungsgruppen finden Sie unter Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe.
16. Wählen Si Create deployment group (Bereitstellungsgruppe erstellen).

Der nächste Schritt besteht in der Vorbereitung einer Revision zur Bereitstellung für die Anwendung und Bereitstellungsgruppe. Detaillierte Anweisungen finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.

Eine Anwendung für eine Amazon ECS-Servicebereitstellung erstellen (Konsole)

Sie können die CodeDeploy Konsole verwenden, um eine Anwendung für eine Amazon ECS-Servicebereitstellung zu erstellen.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie Erste Schritte aus.
3. Wählen Sie auf der Seite Anwendung erstellen die Option Verwenden aus CodeDeploy.
4. Geben Sie unter Application name (Anwendungsname) den Namen Ihrer Anwendung ein.
5. Wählen Sie unter Compute Platform Amazon ECS aus.
6. Wählen Sie Create application aus.
7. Wählen Sie auf Ihrer Anwendungsseite über die Registerkarte Deployment groups (Bereitstellungsgruppen) die Option Create deployment group (Bereitstellungsgruppe erstellen) aus. Weitere Informationen darüber, was Sie zum Erstellen einer Bereitstellungsgruppe für eine Amazon ECS-Bereitstellung benötigen, finden Sie unterBevor Sie mit einer Amazon-ECS-Bereitstellung beginnen.
8. Geben Sie im Feld Deployment group name (Name der Bereitstellungsgruppe) einen Namen ein, der die Bereitstellungsgruppe beschreibt.

> **Note**
>
> Wenn Sie dieselben Einstellungen verwenden möchten, die in einer anderen Bereitstellungsgruppe verwendet wurden (einschließlich Bereitstellungsgruppenname und Bereitstellungskonfiguration), wählen Sie diese Einstellungen auf dieser Seite. Obwohl diese neue Gruppe und die bestehende Gruppe möglicherweise denselben Namen haben, werden sie als separate Bereitstellungsgruppen CodeDeploy behandelt, da jede Gruppe einer separaten Anwendung zugeordnet ist.

9. Wählen Sie unter Servicerolle eine Servicerolle aus, die CodeDeploy Zugriff auf Amazon ECS gewährt. Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.
10. Wählen Sie unter Load Balancer-Name den Namen des Load Balancers aus, der den Datenverkehr an Ihren Amazon ECS-Service weiterleitet.
11. Wählen Sie unter Production Listener Port den Port und das Protokoll für den Listener aus, der Produktionsdatenverkehr für Ihren Amazon ECS-Service bereitstellt.

12. (Optional) Wählen Sie unter Test-Listener-Port den Port und das Protokoll eines Test-Listeners aus, der während der Bereitstellung den Datenverkehr an die Ersatzaufgabe weiterleitet, die in Ihrem Amazon ECS-Service festgelegt wurde. Sie können eine oder mehrere Lambda-Funktionen in der AppSpec Datei angeben, die während des `AfterAllowTestTraffic` Hooks ausgeführt werden. Die Funktionen können Validierungstests ausführen. Wenn ein Validierungstest fehlschlägt, wird ein Bereitstellungs-Rollback ausgelöst. Wenn die Validierungstests erfolgreich sind, wird der nächste Hook im Bereitstellungslebenszyklus, `BeforeAllowTraffic`, ausgelöst. Wenn kein Test-Listener-Port angegeben ist, passiert während des `AfterAllowTestTraffic` Hooks nichts. Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“ für eine Amazon ECS-Bereitstellung.
13. Wählen Sie unter Name der Zielgruppe 1 und Name der Zielgruppe 2 die Zielgruppen aus, die während Ihrer Bereitstellung für die Weiterleitung des Datenverkehrs verwendet werden. CodeDeploy bindet eine Zielgruppe an den ursprünglichen Aufgabensatz Ihres Amazon ECS-Service und die andere an seinen Ersatzaufgabensatz. Weitere Informationen finden Sie unter Zielgruppen für Ihre Application Load Balancer.
14. Wählen Sie Verkehr sofort umleiten oder Geben Sie an, wann der Verkehr umgeleitet werden soll, um zu bestimmen, wann der Verkehr an Ihren aktualisierten Amazon ECS-Service umgeleitet werden soll.

    Wenn Sie „Verkehr sofort umleiten“ wählen, leitet die Bereitstellung den Datenverkehr automatisch um, nachdem der Ersatzaufgabensatz bereitgestellt wurde.

    Wenn Sie „Geben Sie an, wann der Verkehr umgeleitet werden soll“ wählen, wählen Sie die Anzahl der Tage, Stunden und Minuten aus, nach der erfolgreichen Bereitstellung des Ersatz-Tasksets gewartet werden soll. Während dieser Wartezeit werden Validierungstests in den in der AppSpec Datei angegebenen Lambda-Funktionen ausgeführt. Wenn die Wartezeit abläuft, bevor der Verkehr umgeleitet wird, ändert sich der Bereitstellungsstatus auf. `Stopped`
15. Wählen Sie für Kündigung der ursprünglichen Revision die Anzahl der Tage, Stunden und Minuten aus, die nach einer erfolgreichen Bereitstellung gewartet werden soll, bis die ursprüngliche Aufgabe, die in Ihrem Amazon ECS-Service festgelegt wurde, beendet wird.
16. (Optional) Konfigurieren Sie unter Erweitert alle Optionen, die Sie in die Bereitstellung einbeziehen möchten, z. B. Amazon SNS SNS-Benachrichtigungsauslöser, CloudWatch Amazon-Alarme oder automatische Rollbacks.

    Weitere Informationen finden Sie unter Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe.

Erstellen Sie eine Anwendung für ein AWS Lambda Bereitstellung von Funktionen (Konsole)

Sie können die CodeDeploy Konsole verwenden, um eine Anwendung für eine AWS Lambda Funktionsbereitstellung zu erstellen.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie Erste Schritte aus.
3. Wählen Sie auf der Seite Anwendung erstellen die Option Verwenden aus CodeDeploy.
4. Geben Sie unter Application name (Anwendungsname) den Namen Ihrer Anwendung ein.
5. Wählen Sie unter Compute platform (Plattform für die Datenverarbeitung) die Option AWS Lambda aus.
6. Wählen Sie Create application aus.
7. Wählen Sie auf Ihrer Anwendungsseite über die Registerkarte Deployment groups (Bereitstellungsgruppen) die Option Create deployment group (Bereitstellungsgruppe erstellen) aus.
8. Geben Sie im Feld Deployment group name (Name der Bereitstellungsgruppe) einen Namen ein, der die Bereitstellungsgruppe beschreibt.

   > **Note**
   >
   > Wenn Sie dieselben Einstellungen verwenden möchten, die in einer anderen Bereitstellungsgruppe verwendet wurden (einschließlich Bereitstellungsgruppenname und Bereitstellungskonfiguration), wählen Sie diese Einstellungen auf dieser Seite. Obwohl diese neue Bereitstellungsgruppe und die vorhandene Bereitstellungsgruppe möglicherweise denselben Namen haben, werden sie als separate Bereitstellungsgruppen CodeDeploy behandelt, da jede mit einer separaten Anwendung verknüpft ist.

9. Wählen Sie unter Servicerolle eine Servicerolle aus, die CodeDeploy Zugriff auf gewährt AWS Lambda. Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.
10. Wenn Sie eine vordefinierte Bereitstellungskonfiguration verwenden möchten, dann wählen Sie eine unter Deployment configuration (Bereitstellungskonfiguration) aus und fahren Sie mit Schritt 12 fort. Wenn Sie eine benutzerdefinierte Konfiguration erstellen möchten, fahren Sie mit dem nächsten Schritt fort.

    Weitere Informationen zu Bereitstellungskonfigurationen finden Sie unter Bereitstellungskonfigurationen auf einem AWS Lambda Plattform für die Datenverarbeitung.
11. Wählen Sie zum Erstellen einer benutzerdefinierten Konfiguration Create deployment configuration (Bereitstellungskonfiguration erstellen) aus und gehen Sie wie folgt vor:

    a. Geben Sie für Deployment configuration name (Name der Bereitstellungskonfiguration) einen Namen für die Konfiguration ein.

    b. Wählen Sie für Type (Typ) einen Konfigurationstyp aus. Wenn Sie Canary wählen, wird der Datenverkehr in zwei Inkrementschritten verschoben. Wenn Sie Linear wählen, wird der Datenverkehr in gleichen Inkrementschritten mit einer gleichen Anzahl von Minuten zwischen den einzelnen Inkrementschritten verschoben.

    c. Geben Sie für Step einen Prozentsatz zwischen 1 und 99 des zu verschiebenden Datenverkehrs ein. Für den Konfigurationstyp Canary ist dies der Prozentsatz oder Datenverkehr, der im ersten Inkrementschritt verschoben wird. Der verbleibende Datenverkehr wird nach dem gewählten Intervall im zweiten Inkrementschritt verschoben. Für den Konfigurationstyp Linear ist dies der Prozentsatz oder Datenverkehr, der zu Beginn jedes Intervalls verschoben wird.

    d. Geben Sie unter Interval (Intervall) die Anzahl der Minuten ein. Für den Konfigurationstyp Canary ist dies die Anzahl der Minuten zwischen dem ersten und dem zweiten Verschieben des Datenverkehrs. Für den Konfigurationstyp Linear ist dies die Anzahl der Minuten zwischen jeder inkrementellen Verschiebung.

    > **Note**
    >
    > Die maximale Dauer einer AWS Lambda Bereitstellung beträgt zwei Tage oder 2.880 Minuten. Aus diesem Grund ist der maximale Wert, der für Interval für eine Canary-Konfiguration angegeben werden kann, 2.800 Minuten. Der maximale Wert für eine lineare Konfiguration hängt von dem Wert für Step ab. Ist der Schrittprozentsatz für eine lineare Datenverkehrsverschiebung beispielsweise 25

%, gibt es vier Datenverkehrsverschiebungen. Der maximale Intervallwert ist 2.880 dividiert durch 4, d. h. 720 Minuten.

e. Klicken Sie auf Create deployment configuration (Bereitstellungskonfiguration erstellen).

12. (Optional) Konfigurieren Sie unter Erweitert alle Optionen, die Sie in die Bereitstellung einbeziehen möchten, z. B. Amazon SNS SNS-Benachrichtigungsauslöser, CloudWatch Amazon-Alarme oder automatische Rollbacks.

    Weitere Informationen finden Sie unter Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe.

13. Wählen Si Create deployment group (Bereitstellungsgruppe erstellen).

Anwendung erstellen (CLI)

Um eine Anwendung AWS CLI zu erstellen, rufen Sie den Befehl create-application auf und geben Sie dabei einen Namen an, der die Anwendung eindeutig darstellt. (In einem AWS Konto kann ein CodeDeploy Anwendungsname nur einmal pro Region verwendet werden. Sie können einen Anwendungsnamen in verschiedenen Regionen wiederverwenden.)

Nachdem Sie die AWS CLI zum Erstellen einer Anwendung verwendet haben, besteht der nächste Schritt darin, eine Bereitstellungsgruppe zu erstellen, die die Instanzen angibt, für die Revisionen bereitgestellt werden sollen. Detaillierte Anweisungen finden Sie unter Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy.

Der nächste Schritt nach dem Erstellen der Bereitstellungsgruppe ist die Vorbereitung einer Revision für die Bereitstellung der Anwendung und Bereitstellungsgruppe. Detaillierte Anweisungen finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.

Anwendungsdetails anzeigen mit CodeDeploy

Sie können die CodeDeploy Konsole, die oder die CodeDeploy APIs verwenden AWS CLI, um Details zu allen Anwendungen anzuzeigen, die mit Ihrem AWS Konto verknüpft sind.

Themen

- Anwendungsdetails anzeigen (Konsole)
- Anwendungsdetails anzeigen (CLI)

Anwendungsdetails anzeigen (Konsole)

So verwenden Sie die CodeDeploy Konsole zum Anzeigen von Anwendungsdetails:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie Erste Schritte aus.
3. Um zusätzliche Anwendungsdetails anzuzeigen, wählen Sie den Namen der Anwendung in der Liste aus.

Anwendungsdetails anzeigen (CLI)

Rufen Sie AWS CLI den Befehl, den Befehl oder den get-application Befehl auf, um die batch-get-application Anwendungsdetails anzuzeigen. list-applications

Um Details zu einer einzelnen Anwendung anzuzeigen, rufen Sie den Befehl get-application auf und geben Sie dabei den Namen der Anwendung an.

Um Details zu mehreren Anwendungen anzuzeigen, rufen Sie den Befehl batch-get-applications auf und geben Sie dabei mehrere Anwendungsnamen an.

Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.

Erstellen einer Benachrichtigungsregel

Sie können Benachrichtigungsregeln verwenden, um Benutzer zu benachrichtigen, wenn Änderungen an Bereitstellungsanwendungen vorliegen, z. B. erfolgreiche oder fehlgeschlagene Bereitstellungen. Die Benachrichtigungsregeln spezifizieren sowohl die Ereignisse als auch das Amazon SNS SNS-Thema, das zum Senden von Benachrichtigungen verwendet wird. Weitere Informationen finden Sie unter Was sind Benachrichtigungen?.

Sie können die Konsole oder die verwenden AWS CLI , um Benachrichtigungsregeln für AWS CodeDeploy zu erstellen.

So erstellen Sie eine Benachrichtigungsregel (Konsole):

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy/.
2. Wählen Sie Application (Anwendung), und wählen Sie dann eine Anwendung aus, in der Sie Benachrichtigungen hinzufügen möchten.
3. Wählen Sie auf der Anwendungsseite Notify (Benachrichtigung) und dann Create notification rule (Benachrichtigungsregel erstellen) aus. Sie können auch zur Seite Settings (Einstellungen) für die Anwendung gehen und Create notification rule (Benachrichtigungsregel erstellen) auswählen.
4. Geben Sie unter Notification name (Benachrichtigungsname) einen Namen für die Regel ein.
5. Wählen Sie unter Detailtyp die Option Basic aus, wenn Sie möchten, dass nur die Informationen, die Amazon zur Verfügung gestellt wurden, in der Benachrichtigung EventBridge enthalten sind. Wählen Sie Vollständig, wenn Sie Informationen, die Amazon zur Verfügung gestellt wurden, EventBridge und Informationen, die möglicherweise vom CodeDeploy oder vom Notification Manager bereitgestellt wurden, einbeziehen möchten.

   Weitere Informationen finden Sie unter Inhalt und Sicherheit von Benachrichtigungen verstehen.
6. Wählen Sie unter Events that trigger notifications (Ereignisse, die Benachrichtigungen auslösen) die Ereignisse aus, für die Sie Benachrichtigungen senden möchten.

Kategorie	Ereignisse
Bereitstellung	Fehlgeschlagen
	Erfolgreich
	Gestartet

7. Wählen Sie unter Targets (Ziele) die Option Create SNS topic (SNS-Thema erstellen) aus.

> **Note**
>
> Wenn Sie das Thema erstellen, wird die Richtlinie, die das Veröffentlichen von Ereignissen zu dem Thema ermöglicht CodeDeploy , für Sie angewendet. Wenn Sie ein speziell für CodeDeploy Benachrichtigungen erstelltes Thema verwenden, können Sie außerdem sicherstellen, dass Sie nur Benutzer zur Abonnementliste für das Thema

hinzufügen, für die Sie Benachrichtigungen zu dieser Bereitstellungsanwendung erhalten möchten.

Geben Sie hinter dem Präfix codestar-notifications- einen Namen für das Thema ein und wählen Sie anschließend Submit (Absenden) aus.

> **Note**
>
> Wenn Sie ein vorhandenes Amazon-SNS-Thema verwenden möchten, anstatt ein neues zu erstellen, wählen Sie unter Targets (Ziele), dessen ARN aus. Stellen Sie sicher, dass das Thema über die entsprechende Zugriffsrichtlinie verfügt, und dass die Abonnentenliste nur die Benutzer enthält, die Informationen zur Bereitstellungsanwendung anzeigen dürfen. Weitere Informationen finden Sie unter Konfiguration vorhandener Amazon SNS SNS-Themen für Benachrichtigungen und Grundlegendes zu Inhalt und Sicherheit von Benachrichtigungen.

8. Um die Erstellung der Regel abzuschließen, wählen Sie Submit (Absenden) aus.
9. Sie müssen das Amazon SNS SNS-Thema für die Regel abonnieren, bevor sie Benachrichtigungen erhalten können. Weitere Informationen finden Sie unter Amazon SNS SNS-Themen abonnieren, die sich auf Zielgruppen beziehen. Sie können auch die Integration zwischen Benachrichtigungen und Amazon Q Developer in Chat-Anwendungen einrichten, um Benachrichtigungen an Amazon Chime-Chatrooms oder Slack-Kanäle zu senden. Weitere Informationen finden Sie unter Konfiguration der Integration zwischen Benachrichtigungen und Amazon Q Developer in Chat-Anwendungen.

Um eine Benachrichtigungsregel zu erstellen (AWS CLI)

1. Führen Sie in einem Terminal oder einer Eingabeaufforderung den Befehl create-notification rule aus, um das JSON-Skelett zu generieren:

```
aws codestar-notifications create-notification-rule --generate-cli-skeleton
 > rule.json
```

Sie können die Datei beliebig benennen. In diesem Beispiel heißt die Datei *rule.json*.

2. Öffnen Sie die JSON-Datei in einem Klartext-Editor und bearbeiten Sie sie so, dass sie die Ressource, die Ereignistypen und das Amazon SNS SNS-Ziel enthält, die Sie für die Regel

benötigen. Das folgende Beispiel zeigt eine Benachrichtigungsregel, die **MyNotificationRule** nach einer Anwendung benannt ist, die *MyDeploymentApplication* in einem AWS Konto mit der ID benannt ist. *123456789012* Benachrichtigungen werden mit dem vollständigen Detailtyp an ein Amazon SNS SNS-Thema mit dem Namen gesendet*codestar-notifications-MyNotificationTopic*, wenn Bereitstellungen erfolgreich sind:

```
{
    "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"
}
```

Speichern Sie die Datei.

3. Führen Sie unter Verwendung der soeben bearbeiteten Datei am Terminal oder in der Befehlszeile erneut den Befehl create-notification-rule aus, um die Benachrichtigungsregel zu erstellen:

```
aws codestar-notifications create-notification-rule --cli-input-json
 file://rule.json
```

4. Bei Erfolg gibt der Befehl den ARN der Benachrichtigungsregel zurück, der ähnlich wie im Folgenden dargestellt aussieht:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Eine CodeDeploy Anwendung umbenennen

Sie können die APIs AWS CLI oder die CodeDeploy APIs verwenden, um den Namen einer Anwendung zu ändern.

Um eine Liste von Anwendungsnamen anzuzeigen, verwenden Sie den Befehl AWS CLI to call the list-applications.

Hinweise zur Verwendung von, um den Namen einer Anwendung AWS CLI zu ändern, finden Sie unter update-application.

Informationen zur Verwendung der CodeDeploy APIs zum Ändern eines Anwendungsnamens finden Sie unter API_. UpdateApplication

Löschen Sie eine Anwendung in CodeDeploy

Sie können die CodeDeploy Konsole AWS CLI, die oder eine CodeDeploy API-Aktion verwenden, um Anwendungen zu löschen. Hinweise zur Verwendung der CodeDeploy API-Aktion finden Sie unter DeleteApplication.

> **Warning**
>
> Durch das Löschen einer Anwendung werden Informationen zu der Anwendung aus dem CodeDeploy-System entfernt, einschließlich aller zugehörigen Bereitstellungsgruppeninformationen und Bereitstellungsdetails. Durch das Löschen einer für eine EC2/On-Premises Bereitstellung erstellten Anwendung werden weder Anwendungsrevisionen aus Instances noch Revisionen aus Amazon S3 S3-Buckets gelöscht. Durch das Löschen einer für eine EC2/On-Premises Bereitstellung erstellten Anwendung werden keine Amazon EC2 EC2-Instances beendet oder lokale Instances deregistriert. Diese Aktion kann nicht rückgängig gemacht werden.

Themen

- Löschen Sie eine Anwendung (Konsole)
- Löschen Sie eine Anwendung (AWS CLI)

Löschen Sie eine Anwendung (Konsole)

So verwenden Sie die CodeDeploy Konsole, um eine Anwendung zu löschen:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie in der Liste der Anwendungen die Anwendung aus, die Sie löschen möchten.

   Es wird eine Seite mit Details zur Anwendung angezeigt.
4. Wählen Sie oben rechts die Option Anwendung löschen aus.
5. Wenn Sie dazu aufgefordert werden, geben Sie **`delete`** die Eingabetaste ein, um zu bestätigen, dass Sie die Anwendung löschen möchten, und wählen Sie dann Löschen.

Löschen Sie eine Anwendung (AWS CLI)

Um eine Anwendung AWS CLI zu löschen, rufen Sie den Befehl delete-application auf und geben Sie den Namen der Anwendung an. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.

Arbeiten mit Bereitstellungsgruppen in CodeDeploy

Sie können eine oder mehrere Bereitstellungsgruppen für eine CodeDeploy Anwendung angeben. Jede Anwendungsbereitstellung nutzt eine ihrer Bereitstellungsgruppen. Die Bereitstellungsgruppe enthält während der Bereitstellung verwendete Einstellungen und Konfigurationen. Die meisten Einstellungen für Bereitstellungsgruppen hängen von der Rechenplattform ab, die von Ihrer Anwendung verwendet wird. Einige Einstellungen, wie Rollbacks, Trigger und Alarme, können für Bereitstellungsgruppen für jede Computerplattform konfiguriert werden.

Bereitstellungsgruppen in Bereitstellungen auf Amazon-ECS-Datenverarbeitungsplattformen

In einer Amazon ECS-Bereitstellung spezifiziert eine Bereitstellungsgruppe den Amazon ECS-Service, den Load Balancer, den optionalen Test-Listener und zwei Zielgruppen. Es gibt auch an, wann der Datenverkehr an das Ersatz-Task-Set umgeleitet werden soll und wann das ursprüngliche Task-Set und die Amazon ECS-Anwendung nach einer erfolgreichen Bereitstellung beendet werden sollen.

Bereitstellungsgruppen in AWS Lambda Bereitstellungen von Rechenplattformen

In einer AWS Lambda-Bereitstellung definiert eine Bereitstellungsgruppe eine Reihe von CodeDeploy Konfigurationen für future Bereitstellungen einer AWS Lambda Funktion. Die Bereitstellungsgruppe gibt beispielsweise an, wie der Datenverkehr an eine neue Version einer Lambda-Funktion weitergeleitet werden soll. Sie kann außerdem Alarme und Rollbacks angeben. Eine einzelne Bereitstellung in einer AWS Lambda-Bereitstellungsgruppe kann eine oder mehrere Gruppenkonfigurationen überschreiben.

Bereitstellungsgruppen in EC2/On-Premises Compute Platform-Bereitstellungen

In einer EC2/On-Premises Bereitstellung besteht eine Bereitstellungsgruppe aus einzelnen Instanzen, die für eine Bereitstellung vorgesehen sind. Eine Bereitstellungsgruppe enthält individuell markierte Instances, Amazon EC2-Instances in Amazon EC2 Auto Scaling-Gruppen oder beides.

In einer In-Situ-Bereitstellung werden die Instances in der Bereitstellungsgruppe mit der neuesten Anwendungsrevision aktualisiert.

In einer blue/green Bereitstellung wird der Datenverkehr von einer Gruppe von Instances zu einer anderen umgeleitet, indem die ursprünglichen Instances von einem oder mehreren Load Balancern abgemeldet und eine Ersatzgruppe von Instances registriert wird, auf der in der Regel bereits die neueste Anwendungsversion installiert ist.

Sie können einer Anwendung in CodeDeploy mehr als eine Bereitstellungsgruppe zuordnen. Dadurch ist es möglich, eine Anwendungsrevision für verschiedene Gruppen von Instances zu verschiedenen Zeiten bereitzustellen. Sie können beispielsweise eine Bereitstellungsgruppe zum Bereitstellen einer Anwendungsrevision in einer Gruppe von Instances mit dem Tag `Test` verwenden, um die Qualität des Codes zu gewährleisten. Anschließend stellen Sie dieselbe Anwendungsrevision in einer Bereitstellungsgruppe von Instances mit dem Tag `Staging` für eine zusätzliche Verifizierung bereit. Schließlich, wenn Sie bereit sind, die neueste Anwendung für Kunden zu veröffentlichen, stellen Sie diese in einer Bereitstellungsgruppe von Instances mit dem Tag `Production` bereit.

Sie können auch mehrere Tag-Gruppen verwenden, um die Kriterien für die Instances in einer Bereitstellungsgruppe weiter zu verfeinern. Weitere Informationen finden Sie unter Tagging Instances for Deployments.

Wenn Sie die CodeDeploy Konsole verwenden, um eine Anwendung zu erstellen, konfigurieren Sie gleichzeitig ihre erste Bereitstellungsgruppe. Wenn Sie die verwenden, AWS CLI um eine Anwendung zu erstellen, erstellen Sie ihre erste Bereitstellungsgruppe in einem separaten Schritt.

Eine Liste der Bereitstellungsgruppen, die Ihrem AWS Konto bereits zugeordnet sind, finden Sie unterDetails zur Bereitstellungsgruppe anzeigen mit CodeDeploy.

Informationen zu Amazon EC2 EC2-Instance-Tags finden Sie unter Arbeiten mit Tags mithilfe der Konsole. Weitere Informationen über lokale Instances finden Sie unter Working with On-Premises Instances. Informationen zu Amazon EC2 Auto Scaling finden Sie unterIntegration CodeDeploy mit Amazon EC2 Auto Scaling.

Topics

- the section called “Erstellen Sie eine Bereitstellungsgruppe”
- the section called “Details zur Bereitstellungsgruppe anzeigen”
- the section called “Ändern Sie die Einstellungen der Bereitstellungsgruppe”
- the section called “Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe”

- the section called “Löschen Sie eine Bereitstellungsgruppe”

Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy

Sie können die CodeDeploy Konsole, die AWS CLI, die CodeDeploy APIs oder eine CloudFormation Vorlage verwenden, um Bereitstellungsgruppen zu erstellen. Informationen zur Verwendung einer CloudFormation Vorlage zum Erstellen einer Bereitstellungsgruppe finden Sie unterCloudFormation Vorlagen als CodeDeploy Referenz.

Wenn Sie die CodeDeploy Konsole verwenden, um eine Anwendung zu erstellen, konfigurieren Sie gleichzeitig ihre erste Bereitstellungsgruppe. Wenn Sie die verwenden, AWS CLI um eine Anwendung zu erstellen, erstellen Sie ihre erste Bereitstellungsgruppe in einem separaten Schritt.

Beim Erstellen einer Bereitstellungsgruppe müssen Sie eine Servicerolle festlegen. Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.

Themen

- Erstellen Sie eine Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole)
- Erstellen Sie eine Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung (Konsole)
- Eine Bereitstellungsgruppe für eine Amazon ECS-Bereitstellung erstellen (Konsole)
- Richten Sie in Elastic Load Balancing einen Load Balancer für CodeDeploy Amazon EC2 EC2-Bereitstellungen ein
- Einen Load Balancer, Zielgruppen und Listener für CodeDeploy Amazon ECS-Bereitstellungen einrichten
- Erstellen Sie eine Bereitstellungsgruppe (CLI)

Erstellen Sie eine Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole)

So verwenden Sie die CodeDeploy Konsole, um eine Bereitstellungsgruppe für eine direkte Bereitstellung zu erstellen:

> **Warning**
>
> Führen Sie diese Schritte nicht aus, wenn:

- Sie haben Ihre Instances nicht für die Verwendung in den ersten CodeDeploy-Bereitstellung einer Anwendung vorbereitet. Befolgen Sie die Anweisung unter Arbeiten mit Instanzen für CodeDeploy und befolgen Sie anschließend die Schritte dieses Themas, um Ihre Instances einzurichten.
- Sie möchten eine Bereitstellungsgruppe mit einer benutzerdefinierten Bereitstellungskonfiguration erstellen, aber Sie haben die Bereitstellungskonfiguration noch nicht erstellt. Befolgen Sie die Anweisungen unter Create a Deployment Configuration und befolgen Sie anschließend die Schritte dieses Themas.
- Sie haben keine Servicerolle, die mindestens den unter beschriebenen Vertrauensstellungen und Berechtigungen vertraut CodeDeploy . Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy Um eine Servicerolle zu erstellen und zu konfigurieren, befolgen Sie die Anweisungen unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy und befolgen Sie dann die Schritte dieses Themas.
- Sie möchten einen Classic Load Balancer, Application Load Balancer oder Network Load Balancer in Elastic Load Balancing für die direkte Bereitstellung auswählen, haben ihn aber noch nicht erstellt.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Konsole unter. CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie auf der Seite Applications den Namen der Anwendung aus, für die Sie eine Bereitstellungsgruppe erstellen möchten.
4. Wählen Sie auf Ihrer Anwendungsseite über die Registerkarte Deployment groups (Bereitstellungsgruppen) die Option Create deployment group (Bereitstellungsgruppe erstellen) aus.
5. Geben Sie im Feld Deployment group name (Name der Bereitstellungsgruppe) einen Namen ein, der die Bereitstellungsgruppe beschreibt.

> **Note**
>
> Wenn Sie dieselben Einstellungen verwenden möchten, die in einer anderen Bereitstellungsgruppe verwendet wurden (einschließlich des Namens der Bereitstellungsgruppe, der Tags, der Amazon EC2 Auto Scaling Scaling-Gruppennamen oder beides sowie der Bereitstellungskonfiguration), geben Sie diese Einstellungen auf dieser Seite an. Obwohl diese neue Bereitstellungsgruppe und die bestehende Bereitstellungsgruppe denselben Namen haben, werden sie als separate Bereitstellungsgruppen CodeDeploy behandelt, da sie jeweils separaten Anwendungen zugeordnet sind.

6. Wählen Sie im Feld Service role (Servicerolle) eine Servicerolle aus, die CodeDeploy-Zugriff auf Ihre Ziel-Instance gewährt.
7. Wählen Sie unter Bereitstellungstyp die Option aus In-place.
8. Gehen Sie unter Umgebungskonfiguration wie folgt vor:

   a. Wenn Sie Ihre Anwendung in einer Amazon EC2 Auto Scaling Scaling-Gruppe bereitstellen möchten, wählen Sie Amazon EC2 Auto Scaling Scaling-Gruppen und dann den Namen einer Amazon EC2 Auto Scaling Scaling-Gruppe aus, für die Sie Ihre Anwendungsrevision bereitstellen möchten. Wenn neue Amazon EC2-Instances als Teil einer Amazon EC2 Auto Scaling-Gruppe gestartet werden, CodeDeploy kann Ihre Revisionen automatisch für die neuen Instances bereitgestellt werden. Sie können einer Bereitstellungsgruppe bis zu 10 Amazon EC2 Auto Scaling Scaling-Gruppen hinzufügen. Weitere Informationen finden Sie unter Integration CodeDeploy mit Amazon EC2 Auto Scaling.

   b. Wenn Sie Amazon EC2 Auto Scaling Scaling-Gruppen ausgewählt haben, wählen Sie optional Einen Terminierungs-Hook zu Auto Scaling Scaling-Gruppen hinzufügen aus, damit bei der Erstellung oder Aktualisierung der Bereitstellungsgruppe ein Terminierungs-Hook in Ihrer Auto Scaling-Gruppe CodeDeploy installiert wird. Wenn dieser Hook installiert ist, CodeDeploy werden Terminierungsbereitstellungen durchgeführt. Weitere Informationen finden Sie unter Aktivierung von Terminierungsbereitstellungen bei Auto Scaling-Scale-In-Ereignissen.

   c. Wenn Sie Ihre Instances taggen möchten, wählen Sie Amazon EC2 EC2-Instances oder On-premises Instances aus. Geben Sie in den Feldern Schlüssel und Wert die Werte des Schlüssel-Wert-Paares ein, das Sie zum Markieren der Instances verwendet haben. Sie können bis zu 10 Schlüssel-Wert-Paare in einer einzigen Tag-Gruppe markieren.

i. Sie können Platzhalter im Feld Wert verwenden, um alle Instances zu identifizieren, die nach bestimmten Mustern gekennzeichnet sind, z. B. ähnliche Amazon EC2 EC2-Instance-, Kostenstellen- und Gruppennamen usw. Wenn Sie beispielsweise „Name“ im Feld „Schlüssel“ wählen und **`GRP-*a`** in das Feld „Wert“ eingeben, CodeDeploy werden alle Instances identifiziert, die diesem Muster entsprechen, z. B. **`GRP-1aGRP-2a`**, und. **`GRP-XYZ-a`**

ii. Das Feld Value (Wert) unterscheidet Groß- und Kleinschreibung.

iii. Klicken Sie auf das Löschen-Symbol, um ein Schlüssel-Wert-Paar aus der Liste zu entfernen.

Wenn CodeDeploy Instances gefunden werden, die jedem angegebenen Schlüssel-Wert-Paar oder Amazon EC2 Auto Scaling Scaling-Gruppennamen entsprechen, wird die Anzahl der passenden Instances angezeigt. Klicken Sie auf die Zahl, um weitere Informationen zu den Instances zu erhalten.

Wenn Sie die Kriterien für die bereitgestellten Instances verfeinern möchten, wählen Sie Tag-Gruppe hinzufügen, um eine Tag-Gruppe zu erstellen. Sie können bis zu drei Tag-Gruppen mit jeweils bis zu 10 Schlüssel-Wert-Paaren erstellen. Wenn Sie mehrere Tag-Gruppen in einer Bereitstellungsgruppe verwenden, gehören nur Instances, die von allen tag-Gruppen identifiziert werden, zu der Bereitstellungsgruppe. Dies bedeutet, dass eine Instance mindestens einem Tag in jeder der Gruppen entsprechen muss, um zu der Bereitstellungsgruppe zu gehören.

Informationen zur Verwendung von Tag-Gruppen zur Verfeinerung Ihrer Bereitstellungsgruppe finden Sie unter Tagging Instances for Deployments.

9. Geben Sie in der Agentenkonfiguration mit Systems Manager an, wie Sie den CodeDeploy Agenten auf den Instanzen in Ihrer Bereitstellungsgruppe installieren und aktualisieren möchten. Weitere Informationen zum CodeDeploy Agenten finden Sie unter Mit dem CodeDeploy Agenten arbeiten. Weitere Informationen zu Systems Manager finden Sie unter Was ist Systems Manager?

   a. Niemals: Überspringen Sie die Konfiguration der CodeDeploy Installation mit Systems Manager. Der Agent muss auf Instances installiert sein, damit er in Bereitstellungen verwendet werden kann. Wählen Sie diese Option also nur aus, wenn Sie den CodeDeploy-Agenten auf eine andere Weise installieren möchten.

b. Nur einmal: Systems Manager installiert den CodeDeploy Agenten einmal auf jeder Instanz in Ihrer Bereitstellungsgruppe.

c. Jetzt und Updates planen: Systems Manager erstellt eine Verknüpfung mit State Manager, der den CodeDeploy Agenten nach dem von Ihnen konfigurierten Zeitplan installiert. Weitere Informationen zum Statusmanager und Zuordnungen finden Sie unter Info zu Statusmanager.

10. Wählen Sie aus der Liste Deployment configuration (Bereitstellungskonfiguration) eine Bereitstellungskonfiguration, um die Rate zu steuern, mit der Instances bereitgestellt werden (z. B. jeweils nur eine oder mehrere gleichzeitig). Weitere Informationen zu Bereitstellungskonfigurationen finden Sie unter Arbeiten mit Bereitstellungskonfigurationen in CodeDeploy.

11. (Optional) Wählen Sie unter Load Balancer die Option Load Balancing aktivieren aus und wählen Sie dann aus den Listen die Classic Load Balancers, Application Load Balancer Balancer-Zielgruppen und Network Load Balancer Balancer-Zielgruppen aus, um den Datenverkehr zu den Instances während der Bereitstellung zu verwalten. CodeDeploy Sie können bis zu 10 Classic Load Balancer und 10 Zielgruppen auswählen, also insgesamt 20 Elemente. Stellen Sie sicher, dass die Amazon EC2 EC2-Instances, für die Sie die Bereitstellung durchführen möchten, bei den ausgewählten Load Balancers (Classic Load Balancers) oder Zielgruppen (Application Load Balancers und Network Load Balancers) registriert sind.

    Während einer Bereitstellung werden die ursprünglichen Instances von den ausgewählten Load Balancern und Zielgruppen abgemeldet, um zu verhindern, dass während der Bereitstellung Traffic an diese Instances weitergeleitet wird. Wenn die Bereitstellung abgeschlossen ist, wird jede Instanz bei allen ausgewählten Classic Load Balancern und Zielgruppen erneut registriert.

    Weitere Informationen zu Load Balancern für CodeDeploy Bereitstellungen finden Sie unter. Integrating CodeDeploy with Elastic Load Balancing

    > **Warning**
    >
    > Wenn Sie in dieser Bereitstellungsgruppe sowohl Auto Scaling Scaling-Gruppen als auch Elastic Load Balancing-Load Balancer konfigurieren und Load Balancer an die Auto Scaling Scaling-Gruppen anhängen möchten, empfehlen wir, diesen Anhang auszufüllen, bevor Sie die CodeDeploy Bereitstellung aus dieser Bereitstellungsgruppe erstellen. Wenn Sie versuchen, den Anhang nach der Erstellung der Bereitstellung zu

vervollständigen, kann dies dazu führen, dass alle Instances unerwartet von den Load Balancern abgemeldet werden.

12. (Optional) Erweitern Sie Erweitert und konfigurieren Sie alle Optionen, die Sie in die Bereitstellung einbeziehen möchten, z. B. Amazon SNS-Benachrichtigungsauslöser, CloudWatch Amazon-Alarme, Auto Scaling-Optionen oder automatische Rollbacks.

    Weitere Informationen finden Sie unter Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe.

13. Wählen Si Create deployment group (Bereitstellungsgruppe erstellen).

Erstellen Sie eine Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung (Konsole)

So verwenden Sie die CodeDeploy Konsole, um eine Bereitstellungsgruppe für eine blue/green Bereitstellung zu erstellen:

⚠ Warning

Führen Sie diese Schritte nicht aus, wenn:

- Sie haben keine Instances installiert, auf denen der CodeDeploy Agent installiert ist und den Sie während des blue/green Bereitstellungsprozesses ersetzen möchten. Befolgen Sie die Anweisung unter Arbeiten mit Instanzen für CodeDeploy und befolgen Sie anschließend die Schritte dieses Themas, um Ihre Instances einzurichten.
- Sie möchten eine Anwendung mit einer benutzerdefinierten Bereitstellungskonfiguration erstellen, aber Sie haben die Bereitstellungskonfiguration noch nicht erstellt. Befolgen Sie die Anweisungen unter Create a Deployment Configuration und befolgen Sie anschließend die Schritte dieses Themas.
- Sie haben keine Servicerolle, die mindestens den unter beschriebenen Vertrauensstellungen und Berechtigungen vertraut CodeDeploy . Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy Um eine Servicerolle zu erstellen und zu konfigurieren, befolgen Sie die Anweisungen unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy und befolgen Sie dann die Schritte dieses Themas.
- Sie haben in Elastic Load Balancing keinen Classic Load Balancer oder Application Load Balancer für die Registrierung der Instances in Ihrer Ersatzumgebung erstellt. Weitere

Informationen finden Sie unter Richten Sie in Elastic Load Balancing einen Load Balancer für CodeDeploy Amazon EC2 EC2-Bereitstellungen ein.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie auf der Seite Applications den Namen der Anwendung aus, für die Sie eine Bereitstellungsgruppe erstellen möchten.
4. Wählen Sie auf Ihrer Anwendungsseite über die Registerkarte Deployment groups (Bereitstellungsgruppen) die Option Create deployment group (Bereitstellungsgruppe erstellen) aus.
5. Geben Sie im Feld Deployment group name (Name der Bereitstellungsgruppe) einen Namen ein, der die Bereitstellungsgruppe beschreibt.

   > **Note**
   >
   > Wenn Sie dieselben Einstellungen verwenden möchten, die in einer anderen Bereitstellungsgruppe verwendet wurden (einschließlich des Namens der Bereitstellungsgruppe, der Tags, der Amazon EC2 Auto Scaling Scaling-Gruppennamen und der Bereitstellungskonfiguration), wählen Sie diese Einstellungen auf dieser Seite aus. Obwohl diese neue Bereitstellungsgruppe und die vorhandene Bereitstellungsgruppe denselben Namen haben, behandelt CodeDeploy sie als separate Bereitstellungsgruppen, da sie mit separaten Anwendungen verknüpft sind.

6. Wählen Sie im Feld Service role (Servicerolle) eine Servicerolle aus, die CodeDeploy-Zugriff auf Ihre Ziel-Instance gewährt.
7. Wählen Sie unter Bereitstellungstyp die Option Blue/green.
8. Gehen Sie in der Umgebungskonfiguration wie folgt vor:

- Wählen Sie die Methode aus, mit der Sie Instanzen für Ihre Ersatzumgebung bereitstellen möchten. Ihnen stehen folgende Optionen zur Verfügung:
  - Amazon EC2 Auto Scaling Scaling-Gruppe automatisch kopieren: CodeDeploy Erstellt eine Amazon EC2 Auto Scaling Scaling-Gruppe, indem eine von Ihnen angegebene kopiert wird.
  - Manually provision instances: Sie müssen erst eine Bereitstellung erstellen, bevor Sie Instances für Ihre Austauschumgebung angeben können. Sie müssen die Instances erstellen, bevor Sie die Bereitstellung starten. Stattdessen geben Sie hier die Instances an, die Sie hier ersetzen möchten.
- Wenn Sie Amazon EC2 Auto Scaling Scaling-Gruppe automatisch kopieren ausgewählt haben, wählen Sie optional Einen Terminierungs-Hook zu Auto Scaling Scaling-Gruppen hinzufügen aus, damit Sie beim Erstellen oder Aktualisieren der Bereitstellungsgruppe einen Terminierungs-Hook in Ihrer Auto Scaling-Gruppe CodeDeploy installieren können. Wenn dieser Hook installiert ist, CodeDeploy werden Terminierungsbereitstellungen durchgeführt. Weitere Informationen finden Sie unter Aktivierung von Terminierungsbereitstellungen bei Auto Scaling-Scale-In-Ereignissen.

9. Geben Sie in der Agentenkonfiguration mit Systems Manager an, wie Sie den CodeDeploy Agenten auf den Instanzen in Ihrer Bereitstellungsgruppe installieren und aktualisieren möchten. Weitere Informationen zum CodeDeploy Agenten finden Sie unter Mit dem CodeDeploy Agenten arbeiten. Weitere Informationen zu Systems Manager finden Sie unter Was ist Systems Manager?

   a. Niemals: Überspringen Sie die Konfiguration der CodeDeploy Installation mit Systems Manager. Der Agent muss auf Instances installiert sein, damit er in Bereitstellungen verwendet werden kann. Wählen Sie diese Option also nur aus, wenn Sie den CodeDeploy-Agenten auf eine andere Weise installieren möchten.

   b. Nur einmal: Systems Manager installiert den CodeDeploy Agenten einmal auf jeder Instanz in Ihrer Bereitstellungsgruppe.

   c. Jetzt und Updates planen: Systems Manager erstellt eine Verknüpfung mit State Manager, der den CodeDeploy Agenten nach dem von Ihnen konfigurierten Zeitplan installiert. Weitere Informationen zum Statusmanager und Zuordnungen finden Sie unter Info zu Statusmanager.

10. Abhängig von Ihrer Wahl in Schritt 8, führen Sie einen der folgenden Schritte aus:

    - Wenn Sie Amazon EC2 Auto Scaling Scaling-Gruppe automatisch kopieren ausgewählt haben: Wählen Sie in der Amazon EC2 Auto Scaling Scaling-Gruppe den Namen der Amazon

EC2 Auto Scaling Scaling-Gruppe, die Sie als Vorlage für die Amazon EC2 Auto Scaling Scaling-Gruppe verwenden möchten, oder geben Sie ihn ein, die für die Instances in Ihrer Ersatzumgebung erstellt wird. Die Anzahl der aktuell fehlerfreien Instances in der Amazon EC2 Auto Scaling Scaling-Gruppe, die Sie auswählen, wird in Ihrer Ersatzumgebung erstellt.

- Wenn Sie Instances manuell bereitstellen ausgewählt haben: Wählen Sie Amazon EC2 Auto Scaling Scaling-Gruppen, Amazon EC2 Auto Scaling Scaling-Instances oder beide aus, um Instances anzugeben, die zu dieser Bereitstellungsgruppe hinzugefügt werden sollen. Geben Sie Amazon EC2 Auto Scaling-Tagwerte oder Amazon EC2 Auto Scaling Scaling-Gruppennamen ein, um die Instances in Ihrer ursprünglichen Umgebung zu identifizieren (d. h. die Instances, die Sie ersetzen möchten oder auf denen die aktuelle Anwendungsrevision ausgeführt wird).

11. Wählen Sie unter Load Balancer die Option Load Balancing aktivieren aus und wählen Sie dann aus den Listen die Classic Load Balancers, Application Load Balancer Balancer-Zielgruppen und Network Load Balancer Balancer-Zielgruppen aus, bei denen Sie Ihre Amazon EC2 EC2-Ersatz-Instances registrieren möchten. Jede Ersatz-Instance wird bei allen ausgewählten Classic Load Balancern und Zielgruppen registriert. Sie können bis zu 10 Classic Load Balancer und 10 Zielgruppen auswählen, also insgesamt 20 Artikel.

    Der Datenverkehr wird entsprechend den von Ihnen ausgewählten Konfigurationseinstellungen für die Datenverkehrsumleitung und Bereitstellung von den ursprünglichen zu den Ersatz-Instances umgeleitet.

    Weitere Informationen zu Load Balancern für CodeDeploy Bereitstellungen finden Sie unter. Integrating CodeDeploy with Elastic Load Balancing

    > ⚠ Warning
    >
    > Wenn Sie in dieser Bereitstellungsgruppe sowohl Auto Scaling Scaling-Gruppen als auch Elastic Load Balancing-Load Balancer konfigurieren und die Load Balancer Auto Scaling Scaling-Gruppen zuordnen möchten, empfehlen wir, diesen Anhang auszufüllen, bevor Sie die CodeDeploy Bereitstellung aus dieser Bereitstellungsgruppe erstellen. Der Versuch, den Anhang nach der Erstellung des Deployments zu vervollständigen, kann dazu führen, dass alle Instances unerwartet von den Load Balancern abgemeldet werden.

12. Prüfen Sie unter Deployment settings die Standardoptionen für die Umleitung des Datenverkehrs an die Austauschumgebung, welche Bereitstellungskonfiguration Sie für die Bereitstellung

verwenden möchten und wie Instances in der ursprünglichen Umgebung nach der Bereitstellung behandelt werden.

Wenn Sie die Einstellungen ändern möchten, fahren Sie mit dem nächsten Schritt fort. Fahren Sie andernfalls mit Schritt 14 fort.

13. Um die Bereitstellungseinstellungen für die blue/green Bereitstellung zu ändern, wählen Sie eine der folgenden Einstellungen.

Einstellung	Optionen
Traffic rerouting	• Sofortige Umleitung des Datenverkehrs: Sobald Instances in der Ersatzumgebung bereitgestellt und die neueste Anwendungsversion auf ihnen installiert ist, werden sie automatisch bei den angegebenen Load Balancern und Zielgruppen registriert, sodass der Datenverkehr an diese umgeleitet wird. Instances in der ursprünglichen Umgebung werden dann abgemeldet. • Ich entscheide, ob der Datenverkehr umgeleitet werden soll: Instances in der Ersatzumgebung werden nicht bei den angegebenen Load Balancern und Zielgruppen registriert, es sei denn, Sie leiten den Datenverkehr manuell um. Wenn die von Ihnen angegebene Wartezeit abläuft, ohne dass Datenverkehr umgeleitet wurde, wird der Bereitstellungsstatus zu "Stopped" geändert.
Bereitstellungskonfiguration	Wählen Sie die Rate, mit der Instances in der Ersatzumgebung bei den Load Balancern und Zielgruppen registriert werden, z. B. einzeln oder alle gleichzeitig.

Einstellung	Optionen
	Note Nachdem der Datenverkehr erfolgreich zu der Austauchumgebung umgeleitet wurde, werden die Instances in der ursprünglichen Umgebung unabhängig von der gewählten Bereitstellungskonfiguration alle gleichzeitig abgemeldet. Weitere Informationen finden Sie unter Arbeiten mit Bereitstellungskonfigurationen in CodeDeploy.

Einstellung	Optionen

Original instances	• Beenden Sie die ursprünglichen Instances in der Bereitstellungsgruppe: Nachdem der Datenverkehr in die Ersatzumgebung umgeleitet wurde, werden die Instances, die von den Load Balancern und Zielgrupp en abgemeldet wurden, nach der von Ihnen angegebenen Wartezeit beendet. Beachten Sie Folgendes:
	  • Wenn in Ihrer Auto Scaling Scaling-Gruppe der Löschschutz aktiviert ist (`prevent-force-dele tion` oder`prevent-all-deleti on`), deaktiviert die CodeDeploy blue/ green Bereitstellung diesen Schutz für die ursprüngliche (blaue) Auto Scaling Scaling-Gruppe, bevor er bei der Bereinigung der Bereitstellung gelöscht wird. Die Ersatzgruppe (grün) für Auto Scaling behält dieselben Löschschu tzeinstellungen wie die ursprüngliche Gruppe bei.
	  • Lassen Sie die ursprünglichen Instances in der Bereitstellungsgruppe laufen: Nachdem der Datenverkehr in die Ersatzumgebung umgeleitet wurde, werden die Instances, die bei den Load Balancern und Zielgruppen abgemeldet wurden, weiter ausgeführt.

14. (Optional) Konfigurieren Sie unter Erweitert Optionen, die Sie in die Bereitstellung einbeziehen möchten, z. B. Amazon SNS-Benachrichtigungsauslöser, CloudWatch Amazon-Alarme, Auto Scaling-Optionen oder automatische Rollbacks.

    Weitere Informationen zum Festlegen erweiterter Optionen für Bereitstellungsgruppen finden Sie unter Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe.

15. Wählen Si Create deployment group (Bereitstellungsgruppe erstellen).

Eine Bereitstellungsgruppe für eine Amazon ECS-Bereitstellung erstellen (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie in der Liste Anwendungen den Namen der Anwendung aus, die der Bereitstellungsgruppe zugeordnet ist, die Sie bearbeiten möchten.
4. Wählen Sie auf der Anwendungsseite unter Bereitstellungsgruppen den Namen der Bereitstellungsgruppe aus, die Sie bearbeiten möchten.
5. Wählen Sie auf Ihrer Anwendungsseite über die Registerkarte Deployment groups (Bereitstellungsgruppen) die Option Create deployment group (Bereitstellungsgruppe erstellen) aus. Weitere Informationen darüber, was Sie zum Erstellen einer Bereitstellungsgruppe für eine Amazon ECS-Bereitstellung benötigen, finden Sie unterBevor Sie mit einer Amazon-ECS-Bereitstellung beginnen.
6. Geben Sie im Feld Deployment group name (Name der Bereitstellungsgruppe) einen Namen ein, der die Bereitstellungsgruppe beschreibt.

   > **Note**
   >
   > Wenn Sie dieselben Einstellungen verwenden möchten, die in einer anderen Bereitstellungsgruppe verwendet wurden (einschließlich Bereitstellungsgruppenname und Bereitstellungskonfiguration), wählen Sie diese Einstellungen auf dieser Seite. Obwohl diese neue Gruppe und die bestehende Gruppe möglicherweise denselben Namen haben, werden sie als separate Bereitstellungsgruppen CodeDeploy behandelt, da jede Gruppe einer separaten Anwendung zugeordnet ist.

7. Wählen Sie unter Servicerolle eine Servicerolle aus, die CodeDeploy Zugriff auf Amazon ECS gewährt. Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.
8. Wählen Sie unter Load Balancer-Name den Namen des Load Balancers aus, der den Datenverkehr an Ihren Amazon ECS-Service weiterleitet.
9. Wählen Sie unter Production Listener Port den Port und das Protokoll für den Listener aus, der Produktionsdatenverkehr für Ihren Amazon ECS-Service bereitstellt.
10. (Optional) Wählen Sie unter Test-Listener-Port den Port und das Protokoll eines Test-Listeners aus, der während der Bereitstellung den Datenverkehr an die Ersatzaufgabe weiterleitet, die in Ihrem Amazon ECS-Service festgelegt wurde. Sie können eine oder mehrere Lambda-Funktionen in der AppSpec Datei angeben, die während des `AfterAllowTestTraffic` Hooks ausgeführt werden. Die Funktionen können Validierungstests ausführen. Wenn ein Validierungstest fehlschlägt, wird ein Bereitstellungs-Rollback ausgelöst. Wenn die Validierungstests erfolgreich sind, wird der nächste Hook im Bereitstellungslebenszyklus, `BeforeAllowTraffic`, ausgelöst. Wenn kein Test-Listener-Port angegeben ist, passiert während des `AfterAllowTestTraffic` Hooks nichts. Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“ für eine Amazon ECS-Bereitstellung.
11. Wählen Sie unter Name der Zielgruppe 1 und Name der Zielgruppe 2 die Zielgruppen aus, die während Ihrer Bereitstellung für die Weiterleitung des Datenverkehrs verwendet werden. CodeDeploy bindet eine Zielgruppe an den ursprünglichen Aufgabensatz Ihres Amazon ECS-Service und die andere an seinen Ersatzaufgabensatz. Weitere Informationen finden Sie unter Zielgruppen für Ihre Application Load Balancer.
12. Wählen Sie Verkehr sofort umleiten oder Geben Sie an, wann der Verkehr umgeleitet werden soll, um zu bestimmen, wann der Verkehr an Ihren aktualisierten Amazon ECS-Service umgeleitet werden soll.

    Wenn Sie „Verkehr sofort umleiten“ wählen, leitet die Bereitstellung den Datenverkehr automatisch um, nachdem der Ersatzaufgabensatz bereitgestellt wurde.

    Wenn Sie „Geben Sie an, wann der Verkehr umgeleitet werden soll“ wählen, wählen Sie die Anzahl der Tage, Stunden und Minuten aus, nach der erfolgreichen Bereitstellung des Ersatz-Tasksets gewartet werden soll. Während dieser Wartezeit werden Validierungstests in den in der AppSpec Datei angegebenen Lambda-Funktionen ausgeführt. Wenn die Wartezeit abläuft, bevor der Verkehr umgeleitet wird, ändert sich der Bereitstellungsstatus auf. `Stopped`

13. Wählen Sie für Kündigung der ursprünglichen Revision die Anzahl der Tage, Stunden und Minuten aus, die nach einer erfolgreichen Bereitstellung gewartet werden soll, bis die ursprüngliche Aufgabe, die in Ihrem Amazon ECS-Service festgelegt wurde, beendet wird.
14. (Optional) Konfigurieren Sie unter Erweitert alle Optionen, die Sie in die Bereitstellung einbeziehen möchten, z. B. Amazon SNS SNS-Benachrichtigungsauslöser, CloudWatch Amazon-Alarme oder automatische Rollbacks.

    Weitere Informationen finden Sie unter Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe.

Richten Sie in Elastic Load Balancing einen Load Balancer für CodeDeploy Amazon EC2 EC2-Bereitstellungen ein

Bevor Sie eine blue/green Bereitstellung oder eine In-Place-Bereitstellung ausführen, für die Sie einen optionalen Load Balancer in der Bereitstellungsgruppe angeben möchten, müssen Sie mindestens einen Classic Load Balancer, Application Load Balancer oder Network Load Balancer in Elastic Load Balancer erstellt haben. Bei blue/green Bereitstellungen verwenden Sie diesen Load Balancer, um die Instances zu registrieren, aus denen Ihre Ersatzumgebung besteht. Instances in Ihrer ursprünglichen Umgebung können optional mit demselben Load Balancer registriert sein. Bei direkten Bereitstellungen wird der Load Balancer verwendet, um Instances, an denen gerade gearbeitet wird, zu deregistrieren und sie erneut zu registrieren CodeDeploy, wenn die Arbeit abgeschlossen ist.

CodeDeploy unterstützt die blue/green direkte Bereitstellung auf Amazon EC2 EC2-Instances hinter mehreren Load Balancern. Nehmen wir beispielsweise an, Sie haben 200 Amazon EC2 EC2-Instances, von denen 100 bei 2 Classic Load Balancers registriert sind, und weitere 100 davon sind bei 4 Zielgruppen in 2 Application Load Balancers registriert. In diesem Szenario können CodeDeploy Sie In-Place-Bereitstellungen für alle 200 Instances durchführen blue/green , obwohl diese auf 2 Classic Load Balancer, 2 Application Load Balancer und 4 Zielgruppen verteilt sind.

CodeDeploy unterstützt bis zu 10 Classic Load Balancer und 10 Zielgruppen, also insgesamt 20 Artikel.

Um einen oder mehrere Classic Load Balancer zu konfigurieren, folgen Sie den Anweisungen unter Tutorial: Classic Load Balancer erstellen im Benutzerhandbuch für Classic Load Balancers. Beachten Sie Folgendes:

- Wählen Sie in Schritt 2: Definieren des Load Balancers unter Create LB Inside dieselbe VPC aus, die Sie beim Erstellen des Instances verwendet haben.
- Wählen Sie in Schritt 5: EC2-Instances bei Ihrem Load Balancer registrieren die Instances aus, die sich derzeit in Ihrer Bereitstellungsgruppe befinden (In-Place-Bereitstellungen) oder die Sie für Ihre ursprüngliche Umgebung (Bereitstellungen) bestimmt haben. blue/green
- Notieren Sie in Schritt 7: Erstellen und Überprüfen des Load Balancers die DNS-Adresse des Load Balancers.

  Wenn Sie beispielsweise Ihren Load Balancer `my-load-balancer` genannt haben, hat die DNS-Adresse ein Format wie `my-load-balancer-1234567890.us-east-2.elb.amazonaws.com`.

Folgen Sie den Anweisungen in einem der folgenden Themen, um einen oder mehrere Application Load Balancer zu konfigurieren:

- Erstellen Sie einen Application Load Balancer
- Tutorial: Erstellen Sie einen Application Load Balancer mit dem AWS CLI

Folgen Sie den Anweisungen in einem der folgenden Themen, um einen oder mehrere Network Load Balancer zu konfigurieren:

- Einen Network Load Balancer erstellen
- Tutorial: Erstellen Sie einen Network Load Balancer mit dem AWS CLI

Einen Load Balancer, Zielgruppen und Listener für CodeDeploy Amazon ECS-Bereitstellungen einrichten

Bevor Sie eine Bereitstellung mit der Amazon ECS-Rechenplattform ausführen, müssen Sie einen Application Load Balancer oder einen Network Load Balancer, zwei Zielgruppen und einen oder zwei Listener erstellen. In diesem Thema erfahren Sie, wie Sie einen Application Load Balancer erstellen. Weitere Informationen finden Sie unter Bevor Sie mit einer Amazon-ECS-Bereitstellung beginnen.

Eine der Zielgruppen leitet den Traffic zum ursprünglichen Task-Set Ihrer Amazon ECS-Anwendung weiter. Die andere Zielgruppe leitet den Datenverkehr an den neuen Aufgabensatz weiter. CodeDeploy Erstellt während der Bereitstellung einen Ersatzaufgabensatz und leitet den

Datenverkehr vom ursprünglichen Aufgabensatz zum neuen um. CodeDeploy bestimmt, welche Zielgruppe für jeden Tasksatz verwendet wird.

Ein Listener wird von Ihrem Load Balancer verwendet, um Datenverkehr an Ihre Zielgruppen weiterzuleiten. Ein Produktions-Listener ist erforderlich. Sie können optional einen Test-Listener angeben, der Datenverkehr an den neuen Aufgabensatz weiterleitet, während Sie Validierungstests durchführen.

Der Load Balancer muss eine VPC mit zwei öffentlichen Subnetzen in verschiedenen Availability Zones verwenden. Die folgenden Schritte zeigen Ihnen, wie Sie Ihre Standard-VPC bestätigen, einen Amazon EC2 Application Load Balancer erstellen und anschließend zwei Zielgruppen für Ihren Load Balancer erstellen. Weitere Informationen finden Sie unter Zielgruppen für Ihre Netzwerk-Loadbalancer.

Überprüfen Sie Ihre Standard-VPC, öffentliche Subnetze und Sicherheitsgruppe

In diesem Thema wird gezeigt, wie Sie einen Amazon EC2 Application Load Balancer, zwei Zielgruppen und zwei Ports erstellen, die während einer Amazon ECS-Bereitstellung verwendet werden können. Einer der Ports ist optional und wird nur benötigt, wenn Sie Datenverkehr für Validierungstests während Ihrer Bereitstellung an einen Testport leiten.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Überprüfen Sie die zu verwendende Standard-VPC. Wählen Sie im Navigationsbereich Your VPCs (Ihre VPCs) aus. Beachten Sie, für welche VPC Yes (Ja) in der Spalte Default VPC (Standard-VPC) angezeigt wird. Dies ist Ihre Standard-VPC. Sie enthält Standard-Subnetze, die Sie verwenden.
3. Wählen Sie Subnets (Subnetze) aus. Notieren Sie sich die Subnetz-IDs von zwei Subnetzen, für die in der Spalte Standardsubnetz Ja angezeigt wird. Sie verwenden diese IDs, wenn Sie Ihren Load Balancer erstellen.
4. Wählen Sie die einzelnen Subnetze aus und klicken Sie dann auf die Registerkarte Description (Beschreibung). Stellen Sie sicher, dass sich die Subnetze, die Sie verwenden möchten, in verschiedenen Availability Zones befinden.
5. Wählen Sie die Subnetze aus und klicken Sie dann auf die Registerkarte Route Table (Routing-Tabelle). Um zu überprüfen, ob jedes Subnetz, das Sie verwenden möchten, ein öffentliches Subnetz ist, vergewissern Sie sich, dass eine Zeile mit einem Link zu einem Internet-Gateway in der Routing-Tabelle enthalten ist.

6. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
7. Wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
8. Überprüfen Sie, ob die Sicherheitsgruppe, die Sie verwenden möchten, verfügbar ist, und notieren Sie sich die Gruppen-ID (z. B. sg-abcd1234). Sie verwenden diese, wenn Sie Ihren Load Balancer erstellen.

Erstellen Sie einen Amazon EC2 Application Load Balancer, zwei Zielgruppen und Listener (Konsole)

So verwenden Sie die Amazon EC2-Konsole, um einen Amazon EC2 Application Load Balancer zu erstellen:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Klicken Sie im Navigationsbereich auf Load Balancers.
3. Klicken Sie auf Load Balancer erstellen.
4. Wählen Sie Application Load Balancer aus und klicken Sie anschließend auf Create (Erstellen).
5. Geben Sie unter Name den Namen Ihres Load Balancer ein.
6. Wählen Sie für Scheme (Schema) die Option internet-facing (mit dem Internet verbunden) aus.
7. Wählen Sie für IP address type (IP-Adresstyp) die Option ipv4 aus.
8. (Optional) Konfigurieren Sie einen zweiten Listener-Port für Ihren Load Balancer. Sie können Bereitstellungs-Validierungstests mithilfe von Testdatenverkehr ausführen, der an diesen Port geleitet wird.
    a. Wählen Sie unter Load Balancer Protocol (Load Balancer-Protokoll) die Option Add listener (Listener hinzufügen) aus.
    b. Wählen Sie unter Load Balancer Protocol für den zweiten Listener HTTP aus.
    c. Geben Sie unter Load Balancer Port (Load Balancer-Port) „**`8080`**“ ein.
9. Wählen Sie unter Availability Zones, in VPC, die Standard-VPC aus und wählen Sie dann die beiden Standardsubnetze aus, die Sie verwenden möchten.
10. Klicken Sie auf Next: Configure Security Settings (Weiter: Sicherheitseinstellungen konfigurieren).
11. Wählen Sie Next: Configure Security Groups (Weiter: Sicherheitsgruppen konfigurieren) aus.

12. Klicken Sie auf Select an existing security group (Eine vorhandene Sicherheitsgruppe auswählen), wählen Sie die Standardsicherheitsgruppe aus und notieren Sie sich die ID.
13. Wählen Sie Weiter: Routing konfigurieren aus.
14. Wählen Sie in Target group (Zielgruppe) die Option New target group (Neue Zielgruppe) aus und konfigurieren Sie Ihre erste Zielgruppe:
    a. Geben Sie im Feld Name einen Namen für die Zielgruppe ein (z. B. **`target-group-1`**).
    b. Wählen Sie für Target type (Zieltyp) die Option IP aus.
    c. Wählen Sie für Protocol die Option HTTP aus. Geben Sie im Feld Port den Wert **`80`** ein.
    d. Klicken Sie auf Weiter: Ziele registrieren.
15. Wählen Sie Next: Review (Weiter: Prüfen) und anschließend Create role (Rolle erstellen) aus.

So erstellen Sie eine zweite Zielgruppe für Ihren Load Balancer.

1. Nachdem Ihr Load Balancer bereitgestellt wurde, öffnen Sie die Amazon EC2 EC2-Konsole. Wählen Sie im Navigationsbereich Target Groups aus.
2. Wählen Sie Zielgruppe erstellen aus.
3. Geben Sie im Feld Name einen Namen für die Zielgruppe ein (z. B. **`target-group-2`**).
4. Wählen Sie für Target type (Zieltyp) die Option IP aus.
5. Wählen Sie für Protocol die Option HTTP aus. Geben Sie im Feld Port den Wert **`80`** ein.
6. Wählen Sie unter VPC die Standard-VPC aus.
7. Wählen Sie Erstellen aus.

> **Note**
>
> Sie müssen zwei Zielgruppen für Ihren Load Balancer erstellt haben, damit Ihre Amazon ECS-Bereitstellung ausgeführt werden kann. Sie verwenden den ARN einer Ihrer Zielgruppen, wenn Sie Ihren Amazon ECS-Service erstellen. Weitere Informationen finden Sie unter Schritt 4: Amazon ECS-Service erstellen im Amazon ECS-Benutzerhandbuch.

Erstellen Sie einen Amazon EC2 Application Load Balancer, zwei Zielgruppen und Listener (CLI)

So erstellen Sie einen Application Load Balancer mit dem AWS CLI:

1. Verwenden des Befehls create-load-balancer, um einen Application Load Balancer zu erstellen. Geben Sie zwei Subnetze, die sich nicht in derselben Availability Zone befinden, und eine Sicherheitsgruppe an.

   ```
   aws elbv2 create-load-balancer --name bluegreen-alb \
   --subnets subnet-abcd1234 subnet-abcd5678 --security-groups sg-abcd1234 --
   region us-east-1
   ```

   Die Ausgabe enthält den Amazon-Ressourcennamen (ARN) des Load Balancers im folgenden Format:

   ```
   arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
   e5ba62739c16e642
   ```

2. Verwenden Sie den Befehl create-target-group, um Ihre erste Zielgruppe zu erstellen. CodeDeploy leitet den Datenverkehr dieser Zielgruppe an den ursprünglichen oder den neuen Aufgabensatz in Ihrem Service weiter.

   ```
   aws elbv2 create-target-group --name bluegreentarget1 --protocol HTTP --port 80 \
   --target-type ip --vpc-id vpc-abcd1234 --region us-east-1
   ```

   Die Ausgabe enthält den ARN der ersten Zielgruppe im folgenden Format:

   ```
   arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
   bluegreentarget1/209a844cd01825a4
   ```

3. Verwenden Sie den Befehl create-target-group, um Ihre zweite Zielgruppe zu erstellen. CodeDeploy leitet den Datenverkehr der Zielgruppe an den Aufgabensatz weiter, der nicht von Ihrer ersten Zielgruppe bedient wird. Wenn Ihre erste Zielgruppe beispielsweise Datenverkehr an den ursprünglichen Aufgabensatz weiterleitet, leitet diese Zielgruppe den Datenverkehr an den neuen Aufgabensatz weiter.

   ```
   aws elbv2 create-target-group --name bluegreentarget2 --protocol HTTP --port 80 \
   --target-type ip --vpc-id vpc-abcd1234 --region us-east-1
   ```

Die Ausgabe enthält den ARN der zweiten Zielgruppe im folgenden Format:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget2/209a844cd01825a4
```

4. Verwenden Sie den Befehl create-listener, um einen Listener mit einer Standardregel zu erstellen, die Produktionsdatenverkehr an Port 80 weiterleitet.

```
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
```

Die Ausgabe enthält den ARN des Listeners im folgenden Format:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

5. (Optional) Verwenden Sie den Befehl create-listener, um einen zweiten Listener mit einer Standardregel zu erstellen, die Testdatenverkehr an Port 8080 weiterleitet. Sie können Bereitstellungs-Validierungstests mithilfe von Testdatenverkehr ausführen, der an diesen Port geleitet wird.

```
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
```

Die Ausgabe enthält den ARN des Listeners im folgenden Format:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

Erstellen Sie eine Bereitstellungsgruppe (CLI)

Rufen Sie den Befehl create-deployment-group AWS CLI auf und geben Sie Folgendes an, um eine Bereitstellungsgruppe zu erstellen:

- Dem Anwendungsnamen. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.
- Ein Name für die Bereitstellungsgruppe. Eine Bereitstellungsgruppe mit diesem Namen wird für die angegebene Anwendung erstellt. Eine Bereitstellungsgruppe kann nur mit einer Anwendung verbunden werden.
- Informationen zu den Tags, Tag-Gruppen oder Amazon EC2 Auto Scaling Scaling-Gruppennamen, die die Instances identifizieren, die in die Bereitstellungsgruppe aufgenommen werden sollen.
- Der Amazon-Ressourcenname (ARN) der Servicerolle, die es CodeDeploy ermöglicht, bei der Interaktion mit anderen AWS Diensten im Namen Ihres AWS Kontos zu handeln. Informationen zum Abrufen des Servicerollen-ARN finden Sie unter Rufen Sie die Servicerolle ARN (CLI) ab. Weitere Informationen zu Servicerollen finden Sie unter Begriffe und Konzepte für Rollen im IAM-Benutzerhandbuch.
- Informationen über die Art der Bereitstellung, entweder direkt vor Ort oder blue/green, die der Bereitstellungsgruppe zugeordnet werden soll.
- (Optional) Der Name einer vorhandenen Bereitstellungskonfiguration. Eine Liste der Bereitstellungskonfigurationen finden Sie unter View Deployment Configuration Details. Wenn nichts angegeben ist, verwendet CodeDeploy eine Standard-Bereitstellungskonfiguration.
- (Optional) Befehle zum Erstellen eines Triggers, der Benachrichtigungen über Bereitstellungs- und Instance-Ereignisse an diejenigen sendet, die ein Amazon Simple Notification Service-Thema abonniert haben. Weitere Informationen finden Sie unter Monitoring Deployments with Amazon SNS Event Notifications.
- (Optional) Befehle zum Hinzufügen vorhandener CloudWatch Alarme zur Bereitstellungsgruppe, die aktiviert werden, wenn eine in einem Alarm angegebene Metrik einen definierten Schwellenwert unter- oder überschreitet.
- (Optional) Befehle für eine Bereitstellung zum Rollback auf die letzte als funktionierend bekannte Version, wenn eine Bereitstellung fehlschlägt oder ein CloudWatch Alarm aktiviert wird.
- (Optional) Befehle für eine Bereitstellung zum Generieren von Lifecycle-Event-Hooks während eines Auto Scale-In-Ereignisses. Weitere Informationen finden Sie unter So funktioniert Amazon EC2 Auto Scaling mit CodeDeploy.
- Für In-Situ-Bereitstellungen:

  - (Optional) Die Namen der Classic Load Balancers, Application Load Balancers oder Network Load Balancers in Elastic Load Balancing, die den Traffic zu den Instances während der Bereitstellungsprozesse verwalten.
- Für Bereitstellungen: blue/green
  - Konfiguration des blue/green Bereitstellungsprozesses:
    - Wie neue Instances in der Austauschumgebung bereitgestellt werden.
    - Ob Datenverkehr sofort an die Austauschumgebung umgeleitet wird oder eine bestimmte Zeit gewartet wird, bis der Datenverkehr manuell umgeleitet werden kann.
    - Ob Instances in der ursprünglichen Umgebung beendet werden sollen.
  - Die Namen der Classic Load Balancers, Application Load Balancers oder Network Load Balancers in Elastic Load Balancing, die für Instances verwendet werden sollen, die in der Ersatzumgebung registriert sind.

> **Warning**
>
> Wenn Sie in Ihrer Bereitstellungsgruppe sowohl eine Auto Scaling Scaling-Gruppe als auch einen Elastic Load Balancing Load Balancer konfigurieren und den Load Balancer der Auto Scaling Scaling-Gruppe zuordnen möchten, empfehlen wir, diesen Anhang auszufüllen, bevor Sie die CodeDeploy Bereitstellung aus dieser Bereitstellungsgruppe erstellen. Wenn Sie versuchen, den Anhang nach der Erstellung der Bereitstellung zu vervollständigen, kann dies dazu führen, dass alle Instances unerwartet vom Load Balancer abgemeldet werden.

Details zur Bereitstellungsgruppe anzeigen mit CodeDeploy

Sie können die CodeDeploy Konsole, die oder die CodeDeploy APIs verwenden AWS CLI, um Details zu allen Bereitstellungsgruppen anzuzeigen, die einer Anwendung zugeordnet sind.

Themen

- Details zur Bereitstellungsgruppe anzeigen (Konsole)
- Details zur Bereitstellungsgruppe anzeigen (CLI)

Details zur Bereitstellungsgruppe anzeigen (Konsole)

So verwenden Sie die CodeDeploy Konsole zum Anzeigen von Bereitstellungsgruppendetails:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie auf der Seite Applications den mit der Bereitstellungsgruppe verbundenen Anwendungsnamen.

   > **Note**
   >
   > Wenn keine Einträge angezeigt werden, stellen Sie sicher, dass die richtige Region ausgewählt ist. Wählen Sie in der Navigationsleiste in der Regionsauswahl eine der Regionen aus, die unter Region und Endpoints in der aufgeführt sind. Allgemeine AWS-Referenz CodeDeploy wird nur in diesen Regionen unterstützt.

4. Um Details über eine individuelle Bereitstellungsgruppe anzuzeigen, wählen Sie auf der Registerkarte Deployment groups (Bereitstellungsgruppen) den Namen der Bereitstellungsgruppe aus.

Details zur Bereitstellungsgruppe anzeigen (CLI)

Rufen Sie entweder AWS CLI den `get-deployment-group` Befehl oder den Befehl auf, um die Details der Bereitstellungsgruppe anzuzeigen. `list-deployment-groups`

Um Details zu einer einzelnen Bereitstellungsgruppe anzuzeigen, rufen Sie den Befehl get-deployment-group auf und geben Sie Folgendes an:

- Den mit der Bereitstellungsgruppe verbundenen Anwendungsnamen. Rufen Sie den Befehl list-applications auf, um den Namen der Anwendung abzurufen.

- Name der Bereitstellungsgruppe. Rufen Sie den Befehl list-deployment-groups auf, um den Namen der Bereitstellungsgruppe abzurufen.

Um eine Liste der Bereitstellungsgruppennamen anzuzeigen, rufen Sie den Befehl list-deployment-groups auf und geben Sie den Anwendungsnamen an, der den Bereitstellungsgruppen zugeordnet ist. Rufen Sie den Befehl list-applications auf, um den Namen der Anwendung abzurufen.

Ändern Sie die Einstellungen der Bereitstellungsgruppe mit CodeDeploy

Sie können die CodeDeploy Konsole, die oder die CodeDeploy APIs verwenden AWS CLI, um die Einstellungen einer Bereitstellungsgruppe zu ändern.

> **Warning**
>
> Verwenden Sie die folgenden Schritte nicht, wenn Sie möchten, dass die Bereitstellungsgruppe eine noch nicht erstellte benutzerdefinierte Bereitstellungsgruppe verwendet. Befolgen Sie stattdessen die Anweisungen unter Create a Deployment Configuration und kehren Sie dann zu diesem Thema zurück. Verwenden Sie die folgenden Schritte nicht, wenn Sie möchten, dass die Bereitstellungsgruppe eine andere, noch nicht erstellte Servicerolle verwendet. Die Servicerolle muss mindestens den unter beschriebenen Berechtigungen vertrauen CodeDeploy Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy. Um eine Servicerolle mit den richtigen Berechtigungen zu erstellen und zu konfigurieren, befolgen Sie die Anweisungen unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy und kehren Sie dann zu diesem Thema zurück.

Themen

- Einstellungen der Bereitstellungsgruppe ändern (Konsole)
- Einstellungen für Bereitstellungsgruppen ändern (CLI)

Einstellungen der Bereitstellungsgruppe ändern (Konsole)

So verwenden Sie die CodeDeploy Konsole, um die Einstellungen der Bereitstellungsgruppe zu ändern:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie in der Liste der Anwendungen den Namen der Anwendung aus, die der Bereitstellungsgruppe zugeordnet ist, die Sie ändern möchten.

   > **Note**
   >
   > Wenn keine Einträge angezeigt werden, stellen Sie sicher, dass die richtige Region ausgewählt ist. Wählen Sie in der Navigationsleiste in der Regionsauswahl eine der Regionen aus, die unter Region und Endpoints in der aufgeführt sind. Allgemeine AWS-Referenz CodeDeploy wird nur in diesen Regionen unterstützt.

4. Wählen Sie die Registerkarte Deployment groups (Bereitstellungsgruppen) und danach den Namen der Bereitstellungsgruppe aus, die Sie ändern möchten.
5. Wählen Sie auf der Seite „Bereitstellungsgruppe“ die Option Bearbeiten aus.
6. Bearbeiten Sie die Bereitstellungsgruppenoptionen nach Bedarf.

   Weitere Informationen zu Bereitstellungsgruppenkomponenten finden Sie unter Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy.
7. Wählen Sie Änderungen speichern aus.

Einstellungen für Bereitstellungsgruppen ändern (CLI)

Um die Einstellungen der Bereitstellungsgruppe AWS CLI zu ändern, rufen Sie den Befehl update-deployment-group auf und geben Sie Folgendes an:

- Für EC2/On-Premises und AWS Lambda-Bereitstellungen:

- Dem Anwendungsnamen. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.
- Der Name der aktuellen Bereitstellungsgruppe. Rufen Sie den Befehl list-deployment-groups auf, um eine Liste der Bereitstellungsgruppennamen anzuzeigen.
- (Optional) Ein anderer Bereitstellungsgruppenname.
- (Optional) Ein anderer Amazon-Ressourcenname (ARN), der einer Servicerolle entspricht, die es CodeDeploy ermöglicht, bei der Interaktion mit anderen AWS Diensten im Namen Ihres AWS Kontos zu handeln. Informationen zum Abrufen des Servicerollen-ARN finden Sie unter Rufen Sie die Servicerolle ARN (CLI) ab. Weitere Informationen zu Servicerollen finden Sie unter Begriffe und Konzepte für Rollen im IAM-Benutzerhandbuch.
- (Optional) Der Name der Bereitstellungskonfiguration. Eine Liste der Bereitstellungskonfigurationen finden Sie unter View Deployment Configuration Details. (Wenn nicht angegeben, wird eine Standardbereitstellungskonfiguration CodeDeploy verwendet.)
- (Optional) Befehle zum Hinzufügen eines oder mehrerer vorhandener CloudWatch Alarme zur Bereitstellungsgruppe, die aktiviert werden, wenn eine in einem Alarm angegebene Metrik einen definierten Schwellenwert unter- oder überschreitet.
- (Optional) Befehle für eine Bereitstellung zum Rollback auf die letzte als funktionierend bekannte Version, wenn eine Bereitstellung fehlschlägt oder ein CloudWatch Alarm aktiviert wird.
- (Optional) Befehle für eine Bereitstellung zum Generieren von Lifecycle-Event-Hooks während eines Auto Scale-In-Ereignisses. Weitere Informationen finden Sie unter So funktioniert Amazon EC2 Auto Scaling mit CodeDeploy.
- (Optional) Befehle zum Erstellen oder Aktualisieren eines Triggers, der zu einem Thema in Amazon Simple Notification Service veröffentlicht, sodass Abonnenten dieses Themas Benachrichtigungen über Bereitstellungs- und Instance-Ereignisse in dieser Bereitstellungsgruppe erhalten. Weitere Informationen finden Sie unter Monitoring Deployments with Amazon SNS Event Notifications.

- Nur für EC2/On-Premises Bereitstellungen:
  - (Optional) Ersatz-Tags oder Tag-Gruppen zur eindeutigen Identifizierung der Instances, die in die Bereitstellungsgruppe eingeschlossen werden sollen.
  - (Optional) Die Namen der Amazon EC2 Auto Scaling Scaling-Ersatzgruppen, die der Bereitstellungsgruppe hinzugefügt werden sollen.
- Nur für Amazon ECS-Bereitstellungen:
  - Der bereitzustellende Amazon ECS-Service.

- Load Balancer-Informationen, einschließlich Application Load Balancer oder Network Load Balancer, die für eine Amazon ECS-Bereitstellung erforderlichen Zielgruppen sowie Produktions- und optionale Test-Listener-Informationen.

Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe

Bei der Erstellung oder Aktualisierung einer Bereitstellungsgruppe können Sie verschiedene Optionen konfigurieren, die Ihnen mehr Kontrolle und einen umfassenderen Überblick über die Bereitstellungen in der Bereitstellungsgruppe geben.

Auf dieser Seite finden Sie Informationen, anhand derer Sie erweiterte Optionen konfigurieren können, wenn Sie im Rahmen der folgenden Themen mit Bereitstellungsgruppen arbeiten:

- Erstellen Sie eine Anwendung mit CodeDeploy
- Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy
- Ändern Sie die Einstellungen der Bereitstellungsgruppe mit CodeDeploy

Amazon SNS SNS-Benachrichtigungsauslöser: Sie können einer CodeDeploy Bereitstellungsgruppe Auslöser hinzufügen, um Benachrichtigungen über Ereignisse im Zusammenhang mit Bereitstellungen in dieser Bereitstellungsgruppe zu erhalten. Diese Benachrichtigungen werden an Empfänger gesendet, die ein Amazon SNS SNS-Thema abonniert haben, das Sie in die Aktion des Auslösers aufgenommen haben.

Sie müssen das Amazon SNS SNS-Thema, auf das dieser Trigger verweisen soll, bereits eingerichtet haben und CodeDeploy müssen über die Berechtigung verfügen, von dieser Bereitstellungsgruppe aus zu diesem Thema zu veröffentlichen. Falls Sie diese Einrichtungsschritte noch nicht durchgeführt haben, können Sie der Bereitstellungsgruppe auch später noch Auslöser hinzufügen.

Wenn Sie sofort einen Auslöser erstellen möchten, um über Bereitstellungsereignisse in der Bereitstellungsgruppe dieser Anwendung benachrichtigt zu werden, klicken Sie auf Create trigger.

Wenn Ihre Bereitstellung auf einer Amazon EC2-Instance erfolgt, können Sie Benachrichtigungen für Instances erstellen und Benachrichtigungen über Instances erhalten.

Weitere Informationen finden Sie unter Monitoring Deployments with Amazon SNS Event Notifications.

CloudWatch Amazon-Alarme: Sie können einen CloudWatch Alarm erstellen, der eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum beobachtet und auf der Grundlage des Werts der Metrik im Verhältnis zu einem bestimmten Schwellenwert über mehrere Zeiträume hinweg eine oder mehrere Aktionen ausführt. Für eine Amazon EC2-Bereitstellung können Sie einen Alarm für eine Instance oder Amazon EC2 Auto Scaling-Gruppe erstellen, die Sie in Ihren CodeDeploy Vorgängen verwenden. Für eine AWS Lambda- und eine Amazon ECS-Bereitstellung können Sie einen Alarm für Fehler in einer Lambda-Funktion erstellen.

Sie können eine Bereitstellung so konfigurieren, dass sie beendet wird, wenn ein CloudWatch Amazon-Alarm feststellt, dass eine Metrik einen definierten Schwellenwert unter- oder überschritten hat.

Sie müssen den Alarm bereits erstellt haben, CloudWatch bevor Sie ihn einer Bereitstellungsgruppe hinzufügen können.

1. Um der Bereitstellungsgruppe Alarmüberwachung hinzuzufügen, klicken Sie unter Alarms (Alarme) auf Add alarm (Alarm hinzufügen).
2. Geben Sie den Namen eines CloudWatch Alarms ein, den Sie bereits zur Überwachung dieser Bereitstellung eingerichtet haben.

   Sie müssen den CloudWatch Alarm genau so eingeben, wie er in erstellt wurde CloudWatch. Um eine Liste der Alarme anzuzeigen, öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/ und wählen Sie dann ALARM.

Zusätzliche Optionen:

- Wenn Sie die Bereitstellungen ohne Berücksichtigung der hinzugefügten Alarme durchführen möchten, wählen SieIgnore alarm configuration.

  Diese Option ist nützlich, wenn Sie die Alarmüberwachung für eine Bereitstellungsgruppe vorübergehend deaktivieren möchten, ohne später dieselben Alarme erneut hinzufügen zu müssen.
- (Optional) Wenn Sie möchten, dass CodeDeploy Bereitstellungen auch dann fortgesetzt werden, wenn der Alarmstatus nicht von Amazon abgerufen werden kann CloudWatch, wählen Sie Bereitstellungen fortsetzen, auch wenn der Alarmstatus nicht verfügbar ist.

> **Note**
>
> Diese Option entspricht ignorePollAlarmFailure in dem AlarmConfigurationObjekt in der CodeDeploy API.

Weitere Informationen finden Sie unter Überwachung von Bereitstellungen mit CloudWatch Alarmen in CodeDeploy.

Automatische Rollbacks: Sie können konfigurieren, dass für eine Bereitstellungsgruppe oder eine Bereitstellung automatisch ein Rollback durchgeführt wird, wenn eine Bereitstellung fehlschlägt oder ein definierter Überwachungsschwellenwert erreicht wird. In diesem Fall wird die letzte als funktionierend bekannte Version der Anwendungsrevision bereitgestellt. Wenn Sie mithilfe der Konsole eine Anwendung oder eine Bereitstellungsgruppe erstellen oder eine Bereitstellungsgruppe aktualisieren, können Sie optionale Einstellungen für die Bereitstellungsgruppe konfigurieren. Bei der Erstellung einer neuen Bereitstellung können Sie den für die Bereitstellungsgruppe konfigurierten automatischen Rollback auch überschreiben.

- Sie können eine der folgenden beiden Optionen auswählen, damit Bereitstellungen bei einem Fehler auf die letzte als funktionierend bekannte Revision zurückgesetzt werden:
  - Führen Sie ein Rollback durch, wenn eine Bereitstellung fehlschlägt. CodeDeploy wird die letzte als funktionierend bekannte Version als neue Bereitstellung erneut bereitstellen.
  - Roll back when alarm thresholds are met. Wenn Sie dieser Anwendung im vorherigen Schritt einen Alarm hinzugefügt haben, CodeDeploy wird die letzte als funktionierend bekannte Version erneut bereitgestellt, sobald einer oder mehrere der angegebenen Alarme aktiviert sind.

> **Note**
>
> Sollen konfigurierte Rollbacks vorübergehend ignoriert werden, wählen Sie die Option Disable rollbacks aus. Diese Option ist nützlich, wenn Sie automatische Rollbacks vorübergehend deaktivieren möchten, ohne dieselbe Konfiguration später erneut einrichten zu müssen.

> Weitere Informationen finden Sie unter Eine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy.

Automatische Updates für veraltete Instances: Unter bestimmten Umständen CodeDeploy kann eine veraltete Version Ihrer Anwendung auf Ihren Amazon EC2 EC2-Instances bereitgestellt werden. Wenn Ihre EC2-Instances beispielsweise während einer laufenden CodeDeploy Bereitstellung in einer Auto Scaling Scaling-Gruppe (ASG) gestartet werden, erhalten diese Instances die ältere Version Ihrer Anwendung statt der neuesten. Um diese Instances auf den neuesten Stand zu bringen, CodeDeploy wird automatisch eine Folgebereitstellung (unmittelbar nach der ersten) gestartet, um alle veralteten Instances zu aktualisieren. Wenn Sie dieses Standardverhalten ändern möchten, sodass veraltete EC2-Instances in der älteren Version belassen werden, können Sie dies über die CodeDeploy API oder die AWS Command Line Interface (CLI) tun.

Um automatische Updates veralteter Instances über die API zu konfigurieren, fügen `outdatedInstancesStrategy` Sie den Anforderungsparameter in die `CreateDeploymentGroup` Aktion `UpdateDeploymentGroup` oder ein. Einzelheiten finden Sie in der AWS CodeDeploy API-Referenz.

Verwenden Sie einen der folgenden Befehle AWS CLI, um die automatischen Updates über zu konfigurieren:

```
aws deploy update-deployment-group arguments --outdated-instances-strategy
UPDATE|IGNORE
```

Oder...

```
aws deploy create-deployment-group arguments --outdated-instances-strategy
UPDATE|IGNORE
```

... wobei durch die Argumente ersetzt *`arguments`* wird, die für Ihr Deployment erforderlich sind, und *`UPDATE|IGNORE`* wird ersetzt durch entweder `UPDATE` um automatische Updates zu aktivieren oder `IGNORE` um sie zu deaktivieren.

Beispiel:

```
aws deploy update-deployment-group --application-name "MyApp" --current-
deployment-group-name "MyDG" --region us-east-1 --outdated-instances-
strategy IGNORE
```

Einzelheiten zu diesen AWS CLI Befehlen finden Sie in der AWS CLI Befehlsreferenz.

Löschen Sie eine Bereitstellungsgruppe mit CodeDeploy

Sie können die CodeDeploy Konsole, die oder die CodeDeploy APIs verwenden AWS CLI, um Bereitstellungsgruppen zu löschen, die Ihrem AWS Konto zugeordnet sind.

> **Warning**
>
> Wenn Sie eine Bereitstellungsgruppe löschen, werden alle mit dieser Bereitstellungsgruppe verknüpften Details ebenfalls aus gelöscht CodeDeploy. Die in der Bereitstellungsgruppe verwendeten Instances bleiben unverändert. Diese Aktion kann nicht rückgängig gemacht werden.

Themen

- Löschen Sie eine Bereitstellungsgruppe (Konsole)
- Eine Bereitstellungsgruppe löschen (CLI)

Löschen Sie eine Bereitstellungsgruppe (Konsole)

Gehen Sie wie folgt vor, um eine Bereitstellungsgruppe mit der CodeDeploy Konsole zu löschen:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie in der Liste der Anwendungen den Namen der Anwendung aus, die der Bereitstellungsgruppe zugeordnet ist.

4. Wählen Sie auf der Seite Application details (Anwendungsdetails) auf der Registerkarte Deployment groups (Bereitstellungsgruppen) den Namen der Bereitstellungsgruppe aus, die Sie löschen möchten.
5. Klicken Sie auf der Seite Deployment details (Bereitstellungsdetails) auf Delete (Löschen).
6. Geben Sie bei Aufforderung den Namen der Bereitstellungsgruppe ein, um das Löschen zu bestätigen, und klicken Sie auf Delete.

Eine Bereitstellungsgruppe löschen (CLI)

Um eine Bereitstellungsgruppe AWS CLI mit zu löschen, rufen Sie den Befehl delete-deployment-group auf und geben Sie Folgendes an:

- Der Name der Anwendung, die der Bereitstellungsgruppe zugeordnet ist. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.
- Der Name der Bereitstellungsgruppe, die der Anwendung zugeordnet ist. Rufen Sie den Befehl list-deployment-groups auf, um eine Liste der Bereitstellungsgruppennamen anzuzeigen.

Arbeiten mit Anwendungsrevisionen für CodeDeploy

In enthält eine Revision eine Version der Quelldateien CodeDeploy, die auf Ihren Instances bereitgestellt CodeDeploy werden, oder Skripts, die auf Ihren Instances ausgeführt CodeDeploy werden.

Sie planen die Revision, fügen der Revision eine AppSpec Datei hinzu und übertragen die Revision dann auf Amazon S3 oder GitHub. Nachdem Sie die Revision übertragen haben, können Sie sie bereitstellen.

Themen

- Planen Sie eine Überarbeitung für CodeDeploy
- Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy
- Wählen Sie einen CodeDeploy Repository-Typ
- Eine Revision CodeDeploy auf Amazon S3 übertragen (nur EC2/On-Premises Bereitstellungen)
- Revisionsdetails der Anwendung anzeigen mit CodeDeploy
- Registrieren Sie eine Anwendungsrevision in Amazon S3 mit CodeDeploy

Planen Sie eine Überarbeitung für CodeDeploy

Eine gute Planung macht die Bereitstellung von Revisionen viel einfacher.

Bei Bereitstellungen auf einer AWS Lambda- oder Amazon ECS-Rechenplattform entspricht eine Revision der Datei. AppSpec Die folgenden Informationen treffen nicht zu. Weitere Informationen finden Sie unter Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy.

Für Bereitstellungen auf einer EC2/On-Premises Rechenplattform erstellen Sie zunächst ein leeres Stammverzeichnis (Ordner) auf dem Entwicklungscomputer. Hier speichern Sie die Quelldateien (wie z. B. Text- und binäre Dateien, ausführbare Dateien, Pakete usw.), die auf den Instances bereitgestellt werden sollen, oder Skripts, die auf den Instances ausgeführt werden sollen.

Zum Beispiel im `/tmp/` Stammordner unter Linux, macOS oder Unix oder im `c:\temp` Stammordner in 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
```

Der Stammordner sollte auch eine Anwendungsspezifikationsdatei (AppSpec Datei) enthalten, wie hier gezeigt. Weitere Informationen finden Sie unter Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy.

Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy

In diesem Thema wird gezeigt, wie Sie Ihrer Bereitstellung eine AppSpec Datei hinzufügen. Es enthält auch Vorlagen zum Erstellen einer AppSpec Datei für ein AWS Lambda und eine EC2/On-Premises Bereitstellung.

Themen

- Fügen Sie eine AppSpec Datei für eine Amazon ECS-Bereitstellung hinzu
- Fügen Sie eine AppSpec Datei hinzu für AWS Lambda-Bereitstellung
- Fügen Sie eine AppSpec Datei für eine EC2/On-Premises Bereitstellung hinzu

Fügen Sie eine AppSpec Datei für eine Amazon ECS-Bereitstellung hinzu

Für eine Bereitstellung auf einer Amazon ECS-Rechenplattform:

- Die AppSpec Datei spezifiziert die für die Bereitstellung verwendete Amazon ECS-Aufgabendefinition, eine Container-Namen- und Port-Zuordnung, die für die Weiterleitung des Datenverkehrs verwendet werden, sowie optionale Lambda-Funktionen, die nach Ereignissen im Bereitstellungslebenszyklus ausgeführt werden.
- Eine Revision entspricht einer AppSpec Datei.
- Eine AppSpec Datei kann mit JSON oder YAML geschrieben werden.

- Eine AppSpec Datei kann als Textdatei gespeichert oder direkt in eine Konsole eingegeben werden, wenn Sie eine Bereitstellung erstellen. Weitere Informationen finden Sie unter Erstellen Sie eine Amazon ECS Compute Platform-Bereitstellung (Konsole).

Um eine AppSpec Datei zu erstellen

1. Kopieren Sie die JSON- oder YAML-Vorlage in einen Texteditor oder in den AppSpec Editor in der Konsole.
2. Ändern Sie die Vorlage nach Bedarf.
3. Verwenden Sie einen JSON- oder YAML-Validator, um Ihre Datei zu validieren. AppSpec Wenn Sie den AppSpec Editor verwenden, wird die Datei validiert, wenn Sie Bereitstellung erstellen wählen.
4. Wenn Sie einen Texteditor verwenden, speichern Sie die Datei. Wenn Sie die verwenden, AWS CLI um Ihre Bereitstellung zu erstellen, verweisen Sie auf die AppSpec Datei, wenn sie sich auf Ihrer Festplatte oder in einem Amazon S3 S3-Bucket befindet. Wenn Sie die Konsole verwenden, müssen Sie Ihre AppSpec Datei auf Amazon S3 übertragen.

AppSpec YAML-Dateivorlage für eine Amazon ECS-Bereitstellung mit Anweisungen

Im Folgenden finden Sie eine YAML-Vorlage einer AppSpec Datei für eine Amazon ECS-Bereitstellung mit allen verfügbaren Optionen. Weitere Informationen zu Lebenszyklusereignissen für die Verwendung im `hooks`-Abschnitt finden Sie unter AppSpec Abschnitt „Hooks“ für eine Amazon ECS-Bereitstellung.

```
# 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 JSON-Datei für eine Amazon ECS-Bereitstellungsvorlage

Im Folgenden finden Sie eine JSON-Vorlage für eine AppSpec Datei für eine Amazon ECS-Bereitstellung mit allen verfügbaren Optionen. Vorlagenanweisungen finden Sie in den Kommentaren zur YAML-Version im vorherigen Abschnitt. Weitere Informationen zu Lebenszyklusereignissen für die Verwendung im `hooks`-Abschnitt finden Sie unter AppSpec Abschnitt „Hooks“ für eine Amazon ECS-Bereitstellung.

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

Fügen Sie eine AppSpec Datei hinzu für AWS Lambda-Bereitstellung

Für eine Bereitstellung auf einer AWS Lambda-Rechenplattform:

- Die AppSpec Datei enthält Anweisungen zu den Lambda-Funktionen, die bereitgestellt und für die Bereitstellungsvalidierung verwendet werden sollen.
- Eine Revision entspricht einer AppSpec Datei.
- Eine AppSpec Datei kann mit JSON oder YAML geschrieben werden.
- Eine AppSpec Datei kann als Textdatei gespeichert oder beim Erstellen einer Bereitstellung direkt in einen AppSpec Konsoleneditor eingegeben werden. Weitere Informationen finden Sie unter Erstellen Sie ein AWS Bereitstellung der Lambda Compute Platform (Konsole).

Um eine AppSpec Datei zu erstellen:

1. Kopieren Sie die JSON- oder YAML-Vorlage in einen Texteditor oder in den AppSpec Editor in der Konsole.
2. Ändern Sie die Vorlage nach Bedarf.
3. Verwenden Sie einen JSON- oder YAML-Validator, um Ihre Datei zu validieren. AppSpec Wenn Sie den AppSpec Editor verwenden, wird die Datei validiert, wenn Sie Bereitstellung erstellen wählen.
4. Wenn Sie einen Texteditor verwenden, speichern Sie die Datei. Wenn Sie die verwenden, AWS CLI um Ihre Bereitstellung zu erstellen, verweisen Sie auf die AppSpec Datei, wenn sie sich auf Ihrer Festplatte oder in einem Amazon S3 S3-Bucket befindet. Wenn Sie die Konsole verwenden, müssen Sie Ihre AppSpec Datei auf Amazon S3 übertragen.

AppSpec YAML-Dateivorlage für ein AWS Lambda Bereitstellung mit Anweisungen

Weitere Informationen zu Lebenszyklusereignissen für die Verwendung im hooks-Abschnitt finden Sie unter AppSpec Abschnitt „Hooks“ für einen AWS Lambda-Bereitstellung.

```
# 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 JSON-Datei für ein AWS Lambda Bereitstellungsvorlage

Ersetzen Sie in der folgenden Vorlage MyFunction "" durch den Namen Ihrer AWS Lambda Funktion. Ersetzen Sie im optionalen Abschnitt Hooks die Lebenszyklusereignisse durch BeforeAllowTraffic oder AfterAllowTraffic.

Weitere Informationen zu Lebenszyklusereignissen für die Verwendung im Hooks-Abschnitt finden Sie unter AppSpec Abschnitt „Hooks“ für einen AWS Lambda-Bereitstellung.

```
{
  "version": 0.0,
  "Resources": [{
   "MyFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
     "Name": "",
     "Alias": "",
     "CurrentVersion": "",
     "TargetVersion": ""
    }
   }
  }],
  "Hooks": [{
    "LifeCycleEvent": ""
```

```
    }
  ]
}
```

Fügen Sie eine AppSpec Datei für eine EC2/On-Premises Bereitstellung hinzu

Ohne eine AppSpec Datei CodeDeploy können Sie die Quelldateien in Ihrer Anwendungsrevision nicht ihren Zielen zuordnen oder Skripts für Ihre Bereitstellung auf einer EC2/On-Premises Rechenplattform ausführen.

Jede Revision darf nur eine AppSpec Datei enthalten.

Um einer Revision eine AppSpec Datei hinzuzufügen:

1. Kopieren Sie die Vorlage in einen Texteditor.
2. Ändern Sie die Vorlage nach Bedarf.
3. Verwenden Sie einen YAML-Validator, um die Gültigkeit Ihrer Datei zu überprüfen. AppSpec
4. Speichern Sie Datei als `appspec.yml` im Stammverzeichnis der Revision.
5. Führen Sie einen der folgenden Befehle aus, um zu überprüfen, ob Sie Ihre AppSpec Datei im Stammverzeichnis abgelegt haben:

   - Für Linux, macOS oder Unix:

     ```
     find /path/to/root/directory -name appspec.yml
     ```

     Es erfolgt keine Ausgabe, wenn die AppSpec Datei dort nicht gefunden wird.

   - Für Windows:

     ```
     dir path\to\root\directory\appspec.yml
     ```

     Ein Fehler „Datei nicht gefunden“ wird angezeigt, wenn die AppSpec Datei dort nicht gespeichert ist.

6. Übertragen Sie die Revision auf Amazon S3 oder GitHub.

   Detaillierte Anweisungen finden Sie unter Eine Revision CodeDeploy auf Amazon S3 übertragen (nur EC2/On-Premises Bereitstellungen).

AppSpec Dateivorlage für eine EC2/On-Premises Bereitstellung mit Anweisungen

> **Note**
>
> Bereitstellungen auf Windows Server-Instanzen unterstützen das `runas` Element nicht. Wenn Sie es auf Windows Server-Instanzen bereitstellen, nehmen Sie es nicht in Ihre AppSpec Datei auf.

```
# 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:
```

Wählen Sie einen CodeDeploy Repository-Typ

Der Speicherort für Dateien, die von benötigt werden, CodeDeploy wird als Repository bezeichnet. Die Verwendung eines Repositorys hängt davon ab, welche Rechenplattform Ihre Bereitstellung verwendet.

- EC2/On-Premises: Um Ihren Anwendungscode auf einer oder mehreren Instanzen bereitzustellen, muss Ihr Code in einer Archivdatei gebündelt und in einem Repository abgelegt werden, wo CodeDeploy Sie während des Bereitstellungsprozesses darauf zugreifen können. Sie bündeln Ihren bereitstellbaren Inhalt und eine AppSpec Datei in einer Archivdatei und laden sie dann in einen der Repository-Typen hoch, die von unterstützt werden. CodeDeploy
- AWS Lambda und Amazon ECS: Bereitstellungen erfordern eine AppSpec Datei, auf die während einer Bereitstellung auf eine der folgenden Arten zugegriffen werden kann:
  - Aus einem Amazon S3 S3-Bucket.
  - Aus Text, der direkt in den AppSpec Editor in der Konsole eingegeben wurde. Weitere Informationen erhalten Sie unter Erstellen Sie ein AWS Bereitstellung der Lambda Compute Platform (Konsole) und Erstellen Sie eine Amazon ECS Compute Platform-Bereitstellung (Konsole).
  - Wenn Sie den verwenden AWS CLI, können Sie auf eine AppSpec Datei verweisen, die sich auf Ihrer Festplatte oder auf einem Netzlaufwerk befindet. Weitere Informationen erhalten Sie unter Erstellen Sie ein AWS Bereitstellung der Lambda Compute Platform (CLI) und Erstellen Sie eine Amazon ECS Compute Platform-Bereitstellung (CLI).

CodeDeploy unterstützt derzeit die folgenden Repository-Typen:

Repository-Typ	Repository-Details	Unterstützte Datenverarbeitungsplattformen
Amazon S3	Amazon Simple Storage Service (Amazon S3) ist die	Bereitstellungen, die die folgenden Rechenplattformen

	AWS Lösung für sicheren, skalierbaren Objektspeicher. Amazon S3 speichert Daten als Objekte in Buckets. Objekte umfassen eine Datei und optional beliebige Metadaten, die die Datei beschreiben. Um ein Objekt in Amazon S3 zu speichern, laden Sie die Datei in einen Bucket hoch. Beim Hochladen der Datei können Sie Berechtigungen und Metadaten zu diesem Objekt festlegen. Weitere Informationen: • Erstellen Sie einen Bucket in Amazon S3 • Eine Revision CodeDeploy auf Amazon S3 übertragen (nur EC2/On-Premises Bereitstellungen) • Automatische Bereitstellung von Amazon S3 aus mit CodeDeploy	verwenden, können die Revision in einem Amazon S3 S3-Bucket speichern. • EC2/On-Premises • AWS Lambda • Amazon ECS

GitHub	Sie können Ihre Anwendungsrevisionen in GitHubRepositorys speichern. Sie können ein Deployment von einem GitHub Repository aus auslösen, wenn der Quellcode in diesem Repository geändert wird. Weitere Informationen: • Integrieren CodeDeploy mit GitHub • Tutorial: Verwenden Sie CodeDeploy zum Bereitstellen einer Anwendung von GitHub	Nur EC2/On-Premises Bereitstellungen können die Revision in einem GitHub Repository speichern.

Bitbucket	Du kannst Code für Bereitstellungsgruppen von EC2-Instances bereitstellen, indem du die CodeDeploy Pipe in Bitbucket-Pipelines verwendest. Bitbucket Pipelines bietet Funktionen für kontinuierliche Integration und kontinuierliche Bereitstellung (CI/CD), einschließlich Bitbucket-Deployments. Die CodeDeploy Pipe überträgt das Artefakt zunächst in einen von Ihnen angegebenen S3-Bucket und stellt dann das Code-Artefakt aus dem Bucket bereit. Weitere Informationen: • Sieh dir die Pipe für Bitbucket an CodeDeploy	Nur EC2/On-Premises Bereitstellungen können die Revision in einem BitBucket Repository speichern.

> **Note**
>
> Eine AWS Lambda Bereitstellung funktioniert nur mit einem Amazon S3 S3-Repository.

Eine Revision CodeDeploy auf Amazon S3 übertragen (nur EC2/On-Premises Bereitstellungen)

Nachdem Sie Ihre Revision wie unter beschrieben geplant Planen Sie eine Überarbeitung für CodeDeploy und der Revision eine AppSpec Datei hinzugefügt haben, wie unter beschriebenHinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy, sind Sie bereit, die Komponentendateien zu bündeln und die Revision an Amazon S3 zu übertragen. Bei Bereitstellungen auf Amazon EC2 EC2-Instances können CodeDeploy Sie nach dem Push der Revision die Revision von Amazon S3 auf den Instances bereitstellen.

> **Note**
>
> CodeDeploy kann auch verwendet werden, um Versionen bereitzustellen, auf die gepusht wurde. GitHub Weitere Informationen finden Sie in Ihrer GitHub Dokumentation.

Wir nehmen an, dass Sie bereits die Anweisungen unter Erste Schritte mit CodeDeploy ausgeführt haben, um AWS CLI einzurichten. Dies ist besonders wichtig, um den später beschriebenen push-Befehl aufrufen zu können.

Stellen Sie sicher, dass Sie einen Amazon S3 S3-Bucket haben. Folgen Sie den Anweisungen unter Bucket erstellen.

Wenn Ihre Bereitstellung auf Amazon EC2 EC2-Instances erfolgt, muss der Amazon S3 S3-Ziel-Bucket erstellt worden sein oder in derselben Region wie die Ziel-Instances vorhanden sein. Wenn Sie beispielsweise eine Revision für einige Instances in der Region USA Ost (Nord-Virginia) und andere Instances in der Region USA West (Oregon) bereitstellen möchten, benötigen Sie einen Bucket in der Region USA Ost (Nord-Virginia) mit einer Kopie der Revision und einen weiteren Bucket in der Region USA West (Oregon) mit einer weiteren Kopie derselben Revision. In diesem Szenario müssten Sie dann zwei separate Bereitstellungen erstellen, eine in der Region USA Ost (Nord-Virginia) und eine weitere in der Region USA West (Oregon), obwohl die Revision in beiden Regionen und Buckets identisch ist.

Sie müssen über Berechtigungen zum Hochladen in den Amazon S3 S3-Bucket verfügen. Sie können diese Berechtigungen über eine Amazon S3 S3-Bucket-Richtlinie angeben. In der folgenden Amazon S3 S3-Bucket-Richtlinie ermöglicht die Verwendung des Platzhalterzeichens (*) dem AWS Konto `111122223333` beispielsweise das Hochladen von Dateien in ein beliebiges Verzeichnis im Amazon S3 S3-Bucket mit dem Name`amzn-s3-demo-bucket`:

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

```
                ]
            }
        }
    ]
}
```

Informationen zur Anzeige Ihrer AWS Konto-ID finden Sie unter Ihre AWS Konto-ID finden.

Informationen zum Generieren und Anhängen einer Amazon S3 S3-Bucket-Richtlinie finden Sie unter Beispiele für Bucket-Richtlinien.

Der Benutzer, der den push Befehl aufruft, muss mindestens berechtigt sein, die Revision in jeden Amazon S3 S3-Ziel-Bucket hochzuladen. Die folgende Richtlinie ermöglicht es dem Benutzer beispielsweise, Revisionen an einer beliebigen Stelle im Amazon S3 S3-Bucket mit dem Namen `amzn-s3-demo-bucket` hochzuladen:

JSON

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

Informationen zum Erstellen und Anhängen einer IAM-Richtlinie finden Sie unter Mit Richtlinien arbeiten.

Pushen Sie eine Revision mit dem AWS CLI

> **Note**
>
> Der `push` Befehl bündelt Anwendungsartefakte und eine AppSpec Datei zu einer Revision. Das Dateiformat dieser Revision ist eine komprimierte ZIP-Datei. Der Befehl kann nicht mit

einer AWS Lambda- oder Amazon ECS-Bereitstellung verwendet werden, da beide eine Revision erwarten, bei der es sich um eine JSON-formatted YAML-formatted AppSpec OR-Datei handelt.

Rufen Sie zur Bündelung und Übertragung der Revision für eine Bereitstellung den push-Befehl auf. Die Parameter sind:

- --application-name: (Zeichenfolge) erforderlich. Der Name der CodeDeploy-Anwendung, die der Anwendungsrevision zugeordnet werden soll.
- --s3-location: (Zeichenfolge) erforderlich. Informationen über den Speicherort der Anwendungsrevision, die auf Amazon S3 hochgeladen werden soll. Sie müssen einen Amazon S3 S3-Bucket und einen Schlüssel angeben. Der Schlüssel ist der Name der Revision. CodeDeploy komprimiert den Inhalt, bevor er hochgeladen wird. Verwenden Sie dabei das Format `s3://`*`amzn-s3-demo-bucket`*`/`*`your-key.zip`*.
- --ignore-hidden-files oder --no-ignore-hidden-files: (boolesch) optional. Verwenden Sie das `--no-ignore-hidden-files` Flag (Standard), um versteckte Dateien zu bündeln und auf Amazon S3 hochzuladen. Verwenden Sie die `--ignore-hidden-files` Markierung, um versteckte Dateien nicht zu bündeln und auf Amazon S3 hochzuladen.
- --source (Zeichenfolge) optional. Der Speicherort des Inhalts, der bereitgestellt werden soll, und die AppSpec Datei auf dem Entwicklungscomputer, die gezippt und auf Amazon S3 hochgeladen werden soll. Der Speicherort wird als Pfad relativ zum aktuellen Verzeichnis angegeben. Ist der relative Pfad nicht angegeben oder wird für den Pfad ein einzelner Punkt („.“) verwendet, wird das aktuelle Verzeichnis verwendet.
- --description (Zeichenfolge) optional. Ein Kommentar, der die Anwendungsrevision zusammenfasst. Falls nicht angegeben, wird die Standardzeichenfolge „Uploaded by AWS CLI 'time' UTC“ verwendet, wobei 'time' die aktuelle Systemzeit in koordinierter Weltzeit (UTC) ist.

Sie können den verwenden AWS CLI , um eine Revision für eine Amazon EC2 EC2-Bereitstellung zu pushen. Ein Beispiel für einen Push-Befehl sieht folgendermaßen aus:

Unter Linux, macOS oder 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 .
```

In 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 .
```

Der Befehl hat folgende Auswirkungen:

- Verknüpft die gebündelten Dateien mit einer Anwendung namens `WordPress_App`.
- Fügt der Revision eine Beschreibung bei.
- Ignoriert versteckte Dateien.
- Benennt die Revision `WordPressApp.zip` um und überträgt sie in einen Bucket namens `amzn-s3-demo-bucket`.
- Bündelt alle Dateien im Stammverzeichnis in der Revision.

Nachdem der Push erfolgreich war, können Sie die AWS CLI oder die CodeDeploy Konsole verwenden, um die Version von Amazon S3 aus bereitzustellen. Um diese Version bereitzustellen mit AWS CLI:

Unter Linux, macOS oder 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
```

In 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
```

Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy.

Revisionsdetails der Anwendung anzeigen mit CodeDeploy

Sie können die CodeDeploy Konsole, die oder die CodeDeploy APIs verwenden AWS CLI, um Details zu allen Anwendungsrevisionen anzuzeigen, die in Ihrem AWS Konto für eine bestimmte Anwendung registriert sind.

Weitere Informationen zur Registrierung von Revisionen erhalten Sie unter Registrieren Sie eine Anwendungsrevision in Amazon S3 mit CodeDeploy.

Themen

- Details zur Anwendungsversion anzeigen (Konsole)
- Revisionsdetails der Anwendung anzeigen (CLI)

Details zur Anwendungsversion anzeigen (Konsole)

Rufen Sie Anwendungsrevisionsdetails wie folgt ab:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich Deploy (Bereitstellen) und klicken Sie dann auf Applications (Anwendungen).

   > **Note**
   >
   > Wenn keine Einträge angezeigt werden, stellen Sie sicher, dass die richtige Region ausgewählt ist. Wählen Sie in der Navigationsleiste in der Regionsauswahl eine der Regionen aus, die unter Region und Endpoints in der aufgeführt sind. Allgemeine AWS-Referenz CodeDeploy wird nur in diesen Regionen unterstützt.

3. Wählen Sie den Namen der Anwendung, deren Revision Sie anzeigen möchten.

4. Wählen Sie auf der Seite Application details (Anwendungsdetails) die Registerkarte Revisions (Revisionen) und überprüfen Sie die Liste der Revisionen, die für die Anwendung registriert sind. Wählen Sie eine Revision aus und klicken Sie dann auf View details (Details anzeigen).

Revisionsdetails der Anwendung anzeigen (CLI)

Um die AWS CLI zum Anzeigen einer Anwendungsrevision zu verwenden, rufen Sie entweder den get-application-revisionBefehl oder den list-application-revisionsBefehl auf.

> **Note**
>
> Verweise, die GitHub für Bereitstellungen gelten, gelten nur für EC2/On-Premises Bereitstellungen. Versionen für AWS Lambda Bereitstellungen funktionieren nicht mit. GitHub

Um Details zu einer einzelnen Anwendungsrevision anzuzeigen, rufen Sie den Befehl get-application-revision auf und geben Sie Folgendes an:

- Dem Anwendungsnamen. Rufen Sie den Befehl list-applications auf, um den Namen der Anwendung abzurufen.
- Bei einer Revision, die in gespeichert ist GitHub, der GitHub Repository-Name und die ID des Commits, der auf die Anwendungsrevision verweist, die in das Repository übertragen wurde.
- Bei einer in Amazon S3 gespeicherten Revision der Name des Amazon S3 S3-Buckets, der die Revision enthält, der Name und der Dateityp der hochgeladenen Archivdatei sowie optional die Amazon S3 S3-Versionskennung und das ETag der Archivdatei. Wenn die Versions-ID, das ETag oder beide während eines Aufrufs von register-application-revision angegeben wurden, müssen sie hier angegeben werden.

Um Details zu mehreren Anwendungsrevisionen anzuzeigen, rufen Sie den Befehl list-application-revisions auf und geben Sie Folgendes an:

- Dem Anwendungsnamen. Rufen Sie den Befehl list-applications auf, um den Namen der Anwendung abzurufen.
- Optional, um nur Details für Amazon S3 S3-Anwendungsrevisionen anzuzeigen, der Amazon S3 S3-Bucket-Name, der die Revisionen enthält.

- Optional, um nur Details für Amazon S3 S3-Anwendungsrevisionen anzuzeigen, eine Präfixzeichenfolge, um die Suche auf Amazon S3 S3-Anwendungsrevisionen zu beschränken. (Falls nicht angegeben, CodeDeploy werden alle passenden Versionen der Amazon S3 S3-Anwendung aufgelistet.)
- Optional, ob Revisionsdetails nur für Revisionen angezeigt werden sollen, die Zielrevision einer Bereitstellungsgruppe sind (Falls nicht angegeben, CodeDeploy werden alle passenden Versionen aufgelistet.)
- Optional den Spaltennamen und die Reihenfolge, in der die Liste der Revisionsdetails sortiert werden soll (Falls nicht angegeben, CodeDeploy werden die Ergebnisse in einer beliebigen Reihenfolge aufgelistet.)

Sie können alle Revisionen oder nur die in Amazon S3 gespeicherten Revisionen auflisten. Sie können nicht nur die in gespeicherten Revisionen auflisten. GitHub

Registrieren Sie eine Anwendungsrevision in Amazon S3 mit CodeDeploy

Wenn Sie den Push-Befehl bereits aufgerufen haben, um eine Anwendungsrevision an Amazon S3 zu übertragen, müssen Sie die Revision nicht registrieren. Wenn Sie jedoch eine Revision auf andere Weise auf Amazon S3 hochladen und möchten, dass die Revision in der CodeDeploy Konsole oder über die angezeigt wird AWS CLI, gehen Sie wie folgt vor, um die Revision zuerst zu registrieren.

Wenn Sie eine Anwendungsrevision in ein GitHub Repository übertragen haben und möchten, dass die Version in der CodeDeploy Konsole oder über die angezeigt wird AWS CLI, müssen Sie auch die folgenden Schritte ausführen.

Sie können nur die AWS CLI oder die CodeDeploy APIs verwenden, um Anwendungsrevisionen in Amazon S3 oder GitHub zu registrieren.

Themen

- Registrieren Sie eine Revision in Amazon S3 mit CodeDeploy (CLI)
- Eine Revision registrieren GitHub mit CodeDeploy (CLI)

Registrieren Sie eine Revision in Amazon S3 mit CodeDeploy (CLI)

1. Laden Sie die Revision auf Amazon S3 hoch.

2. Rufen Sie den Befehl register-application-revision auf und geben Sie Folgendes an:

   - Dem Anwendungsnamen. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.
   - Informationen über die zu registrierende Revision:
     - Der Name des Amazon S3 S3-Buckets, der die Revision enthält.
     - Name und Dateityp der hochgeladenen Revision. Bei AWS Lambda Bereitstellungen ist die Revision eine in JSON oder YAML geschriebene AppSpec Datei. Bei EC2/On-Premises Bereitstellungen enthält die Revision eine Version der Quelldateien, die auf Ihren Instanzen bereitgestellt CodeDeploy werden, oder Skripts, die auf Ihren Instanzen ausgeführt CodeDeploy werden.

       > **Note**
       >
       > Die Dateiformate tar und komprimierte Tar-Archive (.tar und .tar.gz) werden für Windows Server-Instanzen nicht unterstützt.

     - (Optional) Die Amazon S3 S3-Versions-ID der Revision. (Wenn die Versions-ID nicht angegeben ist, CodeDeploy wird die neueste Version verwendet.)
     - (Optional) Das ETag der Revision. (Wenn das ETag nicht angegeben wird, führt CodeDeploy keine Objektvalidierung durch.)
   - (Optional) Eine beliebige Beschreibung, die Sie der Revision zuordnen möchten.

Informationen über eine Revision in Amazon S3 können in der Befehlszeile angegeben werden, wobei diese Syntax als Teil des register-application-revision Aufrufs verwendet wird. (`version`und `eTag` sind optional.)

Für eine Revisionsdatei für eine EC2/On-Premises Bereitstellung:

```
--s3-location bucket=string,key=string,bundleType=tar|tgz|
zip,version=string,eTag=string
```

Für eine Revisionsdatei für eine AWS Lambda-Bereitstellung:

```
--s3-location bucket=string,key=string,bundleType=JSON|YAML,version=string,eTag=string
```

Eine Revision registrieren GitHub mit CodeDeploy (CLI)

> **Note**
>
> AWS Lambda Bereitstellungen funktionieren nicht mit GitHub.

1. Laden Sie die Revision in Ihr GitHub Repository hoch.
2. Rufen Sie den Befehl register-application-revision auf und geben Sie Folgendes an:
   - Dem Anwendungsnamen. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.
   - Informationen über die zu registrierende Revision:
     - Der GitHub Benutzer- oder Gruppenname, der dem Repository zugewiesen wurde, das die Revision enthält, gefolgt von einem Schrägstrich (/), gefolgt vom Repository-Namen.
     - Die ID des Commits, der sich auf die Revision im Repository bezieht.
   - (Optional) Eine beliebige Beschreibung, die Sie der Revision zuordnen möchten.

Informationen über eine Revision in GitHub können in der Befehlszeile angegeben werden, wobei die folgende Syntax als Teil des register-application-revision Aufrufs verwendet wird:

```
--github-location repository=string,commitId=string
```

Arbeiten mit Bereitstellungen in CodeDeploy

Bei CodeDeploy einer Bereitstellung handelt es sich um den Prozess der Installation von Inhalten auf einer oder mehreren Instanzen und die daran beteiligten Komponenten. Dieser Inhalt kann aus Code, Web- und Konfigurationsdateien, ausführbaren Dateien, Paketen, Skripten usw. bestehen. CodeDeploy stellt Inhalte bereit, die in einem Quell-Repository gespeichert sind, gemäß den von Ihnen angegebenen Konfigurationsregeln.

Wenn Sie die EC2/On-Premises Rechenplattform verwenden, können zwei Bereitstellungen für dieselbe Gruppe von Instanzen gleichzeitig ausgeführt werden.

CodeDeploy bietet zwei Bereitstellungstypen: direkte Bereitstellungen und Bereitstellungen. blue/green

- In-place Bereitstellung: Die Anwendung auf jeder Instanz in der Bereitstellungsgruppe wird gestoppt, die neueste Anwendungsversion wird installiert und die neue Version der Anwendung wird gestartet und validiert. Sie können einen Load Balancer verwenden, sodass jede Instanz während der Bereitstellung deregistriert und nach Abschluss der Bereitstellung wieder betriebsbereit ist. Nur Bereitstellungen, die die EC2/On-Premises Rechenplattform verwenden, können direkte Bereitstellungen verwenden. Weitere Informationen zu In-Situ-Bereitstellungen finden Sie unter Überblick über eine direkte Bereitstellung.
- Blue/green Bereitstellung: Das Verhalten Ihrer Bereitstellung hängt davon ab, welche Rechenplattform Sie verwenden:
  - Blue/green auf einer EC2/On-Premises Rechenplattform: Die Instanzen in einer Bereitstellungsgruppe (der ursprünglichen Umgebung) werden mithilfe der folgenden Schritte durch eine andere Gruppe von Instanzen (die Ersatzumgebung) ersetzt:
    - Instanzen werden für die Ersatzumgebung bereitgestellt.
    - Die neueste Anwendungsversion ist auf den Ersatzinstanzen installiert.
    - Für Aktivitäten wie Anwendungstests und Systemverifizierung fällt eine optionale Wartezeit an.
    - Instances in der Ersatzumgebung werden bei einem oder mehreren Elastic Load Balancing Balancing-Load Balancing-Load Balancern registriert, wodurch der Datenverkehr zu ihnen umgeleitet wird. Instances in der ursprünglichen Umgebung werden deregistriert und können beendet oder für andere Zwecke weiterlaufen.

> **Note**
>
> Wenn Sie eine EC2/On-Premises Rechenplattform verwenden, beachten Sie, dass blue/green Bereitstellungen nur mit Amazon EC2 EC2-Instances funktionieren.

- Blue/green auf einer AWS Lambda oder Amazon ECS-Rechenplattform: Der Datenverkehr wird schrittweise entsprechend einer kanarischen, linearen oder All-in-Once-Bereitstellungskonfiguration verschoben.
- Blue/green Bereitstellungen durch CloudFormation: Der Datenverkehr wird im Rahmen eines Stack-Updates von Ihren aktuellen Ressourcen auf Ihre aktualisierten Ressourcen verlagert. CloudFormation Derzeit werden nur blue/green ECS-Bereitstellungen unterstützt.

Weitere Informationen zu blue/green Bereitstellungen finden Sie unter. Überblick über eine Bereitstellung blue/green

Informationen zur automatischen Bereitstellung von Amazon S3 finden Sie unter Automatisches Bereitstellen von Amazon S3 mithilfe von CodeDeploy.

Themen

- Erstellen Sie eine Bereitstellung mit CodeDeploy
- CodeDeploy Bereitstellungsdetails anzeigen
- Protokolldaten für CodeDeploy EC2/On-Premises Bereitstellungen anzeigen
- Stoppen Sie eine Bereitstellung mit CodeDeploy
- Eine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy
- Stellen Sie eine Anwendung in einer anderen Umgebung bereit AWS Konto
- Verwenden Sie den CodeDeploy Agenten, um ein Bereitstellungspaket auf einem lokalen Computer zu validieren

Erstellen Sie eine Bereitstellung mit CodeDeploy

Sie können die CodeDeploy Konsole, die oder die CodeDeploy APIs verwenden AWS CLI, um eine Bereitstellung zu erstellen, die Anwendungsrevisionen installiert, die Sie bereits auf Amazon S3 übertragen haben, oder, wenn Ihre Bereitstellung auf einer EC2/On-Premises Rechenplattform erfolgt GitHub, auf den Instances in einer Bereitstellungsgruppe.

Der Prozess zur Erstellung einer Bereitstellung hängt von der Rechenplattform ab, die von Ihrer Bereitstellung verwendet wird.

Themen

- Voraussetzungen für die Bereitstellung
- Erstellen Sie eine Amazon ECS Compute Platform-Bereitstellung (Konsole)
- Erstellen Sie ein AWS Bereitstellung der Lambda Compute Platform (Konsole)
- Erstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (Konsole)
- Erstellen Sie eine Amazon ECS Compute Platform-Bereitstellung (CLI)
- Erstellen Sie ein AWS Bereitstellung der Lambda Compute Platform (CLI)
- Erstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (CLI)
- Erstellen Sie eine Amazon blue/green ECS-Bereitstellung über CloudFormation

Voraussetzungen für die Bereitstellung

Stellen Sie sicher, dass vor Beginn einer Bereitstellung die folgenden Schritte ausgeführt werden.

Voraussetzungen für die Bereitstellung auf einem AWS Lambda Plattform für die Datenverarbeitung

- Erstellen Sie eine Anwendung, die mindestens eine Bereitstellungsgruppe enthält. Weitere Informationen finden Sie unter Erstellen Sie eine Anwendung mit CodeDeploy und Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy.
- Bereiten Sie die Anwendungsrevision vor, die auch als AppSpec Datei bezeichnet wird und die Lambda-Funktionsversion angibt, die Sie bereitstellen möchten. In der AppSpec Datei können auch Lambda-Funktionen zur Validierung Ihrer Bereitstellung angegeben werden. Weitere Informationen finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.
- Wenn Sie eine benutzerdefinierte Bereitstellungskonfiguration für Ihre Bereitstellung verwenden möchten, erstellen Sie sie, bevor Sie den Bereitstellungsprozess starten. Weitere Informationen finden Sie unter Create a Deployment Configuration.

Voraussetzungen für die Bereitstellung auf einer EC2/on-premises Rechenplattform

- Für eine In-Situ-Bereitstellung erstellen oder konfigurieren Sie die Instances, auf denen Sie bereitstellen möchten. Weitere Informationen finden Sie unter Arbeiten mit Instanzen für

CodeDeploy. Für eine blue/green Bereitstellung haben Sie entweder eine bestehende Amazon EC2 Auto Scaling Scaling-Gruppe, die Sie als Vorlage für Ihre Ersatzumgebung verwenden können, oder Sie haben eine oder mehrere Instances oder Amazon EC2 Auto Scaling Scaling-Gruppen, die Sie als Ihre ursprüngliche Umgebung angeben. Weitere Informationen erhalten Sie unter Tutorial: Wird verwendet CodeDeploy , um eine Anwendung für eine Auto Scaling Scaling-Gruppe bereitzustellen und Integration CodeDeploy mit Amazon EC2 Auto Scaling.

- Erstellen Sie eine Anwendung, die mindestens eine Bereitstellungsgruppe enthält. Weitere Informationen finden Sie unter Erstellen Sie eine Anwendung mit CodeDeploy und Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy.
- Bereiten Sie die Anwendungsrevision vor, die Sie für die einzelnen Instances in der Bereitstellungsgruppe bereitstellen möchten. Weitere Informationen finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.
- Wenn Sie eine benutzerdefinierte Bereitstellungskonfiguration für Ihre Bereitstellung verwenden möchten, erstellen Sie sie, bevor Sie den Bereitstellungsprozess starten. Weitere Informationen finden Sie unter Create a Deployment Configuration.
- Wenn Sie Ihre Anwendungsrevision von einem Amazon S3 S3-Bucket aus bereitstellen, befindet sich der Bucket in derselben AWS Region wie die Instances in Ihrer Bereitstellungsgruppe.
- Wenn Sie Ihre Anwendungsversion von einem Amazon S3 S3-Bucket aus bereitstellen, wurde eine Amazon S3 S3-Bucket-Richtlinie auf den Bucket angewendet. Diese Richtlinie gewährt Ihren Instances die erforderlichen Berechtigungen zum Herunterladen der Anwendungsrevision.

  Die folgende Amazon S3 S3-Bucket-Richtlinie ermöglicht beispielsweise, dass jede Amazon EC2 EC2-Instance mit einem angehängten IAM-Instance-Profil, das den ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` enthält, von einer beliebigen Stelle im Amazon S3 S3-Bucket mit dem Namen heruntergeladen werden kann: `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"
```

```
                ]
            }
        }
    ]
}
```

Die folgende Amazon S3 S3-Bucket-Richtlinie ermöglicht es jeder lokalen Instance mit einem zugehörigen IAM-Benutzer, der den ARN `arn:aws:iam::444455556666:user/CodeDeployUser` enthält, von einer beliebigen Stelle im Amazon S3 S3-Bucket mit dem Namen herunterzuladen: `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"
                ]
            }
        }
    ]
}
```

Informationen zum Generieren und Anhängen einer Amazon S3 S3-Bucket-Richtlinie finden Sie unter Beispiele für Bucket-Richtlinien.

- Wenn Sie eine blue/green Bereitstellung erstellen oder einen optionalen Classic Load Balancer, Application Load Balancer oder Network Load Balancer in der Bereitstellungsgruppe für eine direkte Bereitstellung angegeben haben, haben Sie mit Amazon VPC eine VPC erstellt, die mindestens zwei Subnetze enthält. (CodeDeploy verwendet Elastic Load Balancing, was erfordert, dass sich alle Instances in einer Load Balancer-Gruppe in einer einzigen VPC befinden.)

  Wenn Sie noch keine VPC erstellt haben, finden Sie weitere Informationen im Amazon VPC Getting Started Guide.

- Wenn Sie ein blue/green Deployment erstellen, haben Sie mindestens einen Classic Load Balancer, Application Load Balancer oder Network Load Balancer in Elastic Load Balancing konfiguriert und damit die Instances registriert, aus denen Ihre ursprüngliche Umgebung besteht.

  > **Note**
  >
  > Die Instances in Ihrer Austauschumgebung werden später im Load Balancer registriert.

  Weitere Informationen zur Konfiguration eines Load Balancers finden Sie unter und. Richten Sie in Elastic Load Balancing einen Load Balancer für CodeDeploy Amazon EC2 EC2-Bereitstellungen ein Einen Load Balancer, Zielgruppen und Listener für CodeDeploy Amazon ECS-Bereitstellungen einrichten

Bereitstellungsvoraussetzungen für eine blue/green Bereitstellung über CloudFormation

- Ihre Vorlage muss keine Ressourcen für eine CodeDeploy Anwendung oder Bereitstellungsgruppe modellieren.
- Ihre Vorlage muss Ressourcen für eine VPC enthalten, die Amazon VPC verwendet und mindestens zwei Subnetze enthält.
- Ihre Vorlage muss Ressourcen für einen oder mehrere Classic Load Balancer, Application Load Balancer oder Network Load Balancer in Elastic Load Balancing enthalten, die verwendet werden, um den Traffic an Ihre Zielgruppen weiterzuleiten.

Erstellen Sie eine Amazon ECS Compute Platform-Bereitstellung (Konsole)

In diesem Thema erfahren Sie, wie Sie einen Amazon ECS-Service mithilfe der Konsole bereitstellen. Weitere Informationen erhalten Sie unter Tutorial: Bereitstellen einer Anwendung in Amazon ECS und Tutorial: Bereitstellen eines Amazon ECS-Service mit einem Validierungstest.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Sie eine Anwendung bereitstellen möchten, erweitern Sie im Navigationsbereich Deploy (Bereitstellen) und wählen Sie Applications (Anwendungen) aus. Klicken Sie auf den Namen der Anwendung, die Sie bereitstellen möchten. Stellen Sie sicher, dass die Spalte Compute Platform für Ihre Anwendung Amazon ECS lautet.
   - Wenn Sie eine Bereitstellung erneut bereitstellen möchten, erweitern Sie im Navigationsbereich Deploy (Bereitstellen) und wählen Sie Deployments (Bereitstellungen) aus. Wählen Sie in der Spalte Application (Anwendung) die Bereitstellung aus, die Sie erneut bereitstellen möchten, sowie den Namen ihrer Anwendung. Stellen Sie sicher, dass die Spalte Rechenplattform für Ihre Bereitstellung Amazon ECS lautet.

3. Klicken Sie auf der Registerkarte Deployments (Bereitstellungen) auf Create deployment (Bereitstellung erstellen).

> **Note**
>
> Ihre Anwendung muss über eine Bereitstellungsgruppe verfügen, bevor sie bereitgestellt werden kann. Wenn Ihre Anwendung keine Bereitstellungsgruppe hat, wählen Sie auf der Registerkarte Bereitstellungsgruppen die Option Bereitstellungsgruppe erstellen aus. Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy.

4. Wählen Sie unter Deployment group (Bereitstellungsgruppe) eine Bereitstellungsgruppe für diese Bereitstellung aus.

5. Wählen Sie neben Revision location (Revisionsort) aus, wo sich Ihre Revision befindet:

   - Meine Anwendung ist in Amazon S3 gespeichert. Weitere Informationen finden Sie unterGeben Sie Informationen zu einer Revision an, die in einem Amazon S3 S3-Bucket gespeichert ist, und kehren Sie dann zu Schritt 6 zurück.
   - AppSpec Editor verwenden — Wählen Sie entweder JSON oder YAML aus und geben Sie dann Ihre AppSpec Datei in den Editor ein. Sie können die AppSpec Datei speichern, indem

Sie Als Textdatei speichern wählen. Wenn Sie nach diesen Schritten Deploy (Bereitstellen) auswählen, wird ein Fehler ausgegeben, wenn Ihr JSON oder YAML nicht gültig ist. Weitere Informationen zum Erstellen einer AppSpec Datei finden Sie unterHinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy.

6. (Optional) Geben Sie im Feld Deployment description (Bereitstellungsbeschreibung) eine Beschreibung für diese Bereitstellung ein.
7. (Optional) Sie können unter Rollback configuration overrides andere automatische Rollback-Optionen für diese Bereitstellung festlegen, als für die Bereitstellungsgruppe angegeben wurden.

   Hinweise zu Rollbacks in finden Sie CodeDeploy unter Neubereitstellungen und Bereitstellungs-Rollbacks undEine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy.

   Wählen Sie eine der folgenden Optionen aus:

   - Rollback bei fehlgeschlagener Bereitstellung — CodeDeploy stellt die letzte als funktionierend bekannte Version erneut als neue Bereitstellung bereit.
   - Rollback bei Erreichen der Alarmschwellenwerte — Wenn der Bereitstellungsgruppe Alarme hinzugefügt wurden, wird die letzte als funktionierend bekannte Version CodeDeploy erneut bereitgestellt, wenn einer oder mehrere der angegebenen Alarme aktiviert sind.
   - Rollbacks deaktivieren — Führen Sie für diese Bereitstellung keine Rollbacks durch.
8. Wählen Sie Create deployment.

   Informationen zum Verfolgen des Status Ihrer Bereitstellung finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen.

Erstellen Sie ein AWS Bereitstellung der Lambda Compute Platform (Konsole)

In diesem Thema erfahren Sie, wie Sie eine Lambda-Funktion mithilfe der Konsole bereitstellen.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Sie eine Anwendung bereitstellen möchten, erweitern Sie im Navigationsbereich Deploy (Bereitstellen) und wählen Sie Applications (Anwendungen) aus. Klicken Sie auf den Namen der Anwendung, die Sie bereitstellen möchten. Stellen Sie sicher, dass die Spalte Compute Platform für Ihre Anwendung AWS Lambda lautet.
   - Wenn Sie eine Bereitstellung erneut bereitstellen möchten, erweitern Sie im Navigationsbereich Deploy (Bereitstellen) und wählen Sie Deployments (Bereitstellungen) aus. Wählen Sie in der Spalte Application (Anwendung) die Bereitstellung aus, die Sie erneut bereitstellen möchten, sowie den Namen ihrer Anwendung. Stellen Sie sicher, dass die Spalte Compute Platform für Ihre Bereitstellung AWS Lambda lautet.

3. Klicken Sie auf der Registerkarte Deployments (Bereitstellungen) auf Create deployment (Bereitstellung erstellen).

> **Note**
>
> Ihre Anwendung muss über eine Bereitstellungsgruppe verfügen, bevor sie bereitgestellt werden kann. Wenn Ihre Anwendung keine Bereitstellungsgruppe hat, wählen Sie auf der Registerkarte Bereitstellungsgruppen die Option Bereitstellungsgruppe erstellen aus. Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy.

4. Wählen Sie unter Deployment group (Bereitstellungsgruppe) eine Bereitstellungsgruppe für diese Bereitstellung aus.

5. Wählen Sie neben Revision location (Revisionsort) aus, wo sich Ihre Revision befindet:

   - Meine Anwendung ist in Amazon S3 gespeichert. Weitere Informationen finden Sie unterGeben Sie Informationen zu einer Revision an, die in einem Amazon S3 S3-Bucket gespeichert ist, und kehren Sie dann zu Schritt 6 zurück.
   - AppSpec Editor verwenden — Wählen Sie entweder JSON oder YAML aus und geben Sie dann Ihre AppSpec Datei in den Editor ein. Sie können die AppSpec Datei speichern, indem

Sie Als Textdatei speichern wählen. Wenn Sie nach diesen Schritten Deploy (Bereitstellen) auswählen, wird ein Fehler ausgegeben, wenn Ihr JSON oder YAML nicht gültig ist. Weitere Informationen zum Erstellen einer AppSpec Datei finden Sie unterHinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy.

6. (Optional) Geben Sie im Feld Deployment description (Bereitstellungsbeschreibung) eine Beschreibung für diese Bereitstellung ein.
7. (Optional) Erweitern Sie Deployment-Gruppenüberschreibungen, um eine Bereitstellungskonfiguration auszuwählen, mit der Sie steuern können, wie der Datenverkehr auf die Lambda-Funktionsversion umgeleitet wird, die sich von der in der Bereitstellungsgruppe angegebenen unterscheidet.

   Weitere Informationen finden Sie unter Bereitstellungskonfigurationen auf einem AWS Lambda Plattform für die Datenverarbeitung.
8. (Optional) Sie können unter Rollback configuration overrides andere automatische Rollback-Optionen für diese Bereitstellung festlegen, als für die Bereitstellungsgruppe angegeben wurden.

   Informationen zu Rollbacks in finden Sie unter CodeDeploy und. Neubereitstellungen und Bereitstellungs-Rollbacks Eine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy

   Wählen Sie eine der folgenden Optionen aus:

   - Rollback bei fehlgeschlagener Bereitstellung — CodeDeploy stellt die letzte als funktionierend bekannte Version erneut als neue Bereitstellung bereit.
   - Rollback bei Erreichen der Alarmschwellenwerte — Wenn der Bereitstellungsgruppe Alarme hinzugefügt wurden, wird die letzte als funktionierend bekannte Version CodeDeploy erneut bereitgestellt, wenn einer oder mehrere der angegebenen Alarme aktiviert sind.
   - Rollbacks deaktivieren — Führen Sie für diese Bereitstellung keine Rollbacks durch.
9. Wählen Sie Create deployment.

   Informationen zum Verfolgen des Status Ihrer Bereitstellung finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen.

Erstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (Konsole)

In diesem Thema erfahren Sie, wie Sie mithilfe der Konsole eine Anwendung auf einem Amazon EC2- oder lokalen Server bereitstellen.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Sie eine Anwendung bereitstellen möchten, erweitern Sie im Navigationsbereich Deploy (Bereitstellen) und wählen Sie Applications (Anwendungen) aus. Klicken Sie auf den Namen der Anwendung, die Sie bereitstellen möchten. Stellen Sie sicher, dass die Spalte Compute platform (Plattform für die Datenverarbeitung) für Ihre Anwendung EC2/On-Premises lautet.
   - Wenn Sie eine Bereitstellung erneut bereitstellen möchten, erweitern Sie im Navigationsbereich Deploy (Bereitstellen) und wählen Sie Deployments (Bereitstellungen) aus. Wählen Sie in der Spalte Application (Anwendung) die Bereitstellung aus, die Sie erneut bereitstellen möchten, sowie den Namen ihrer Anwendung aus. Stellen Sie sicher, dass die Spalte Compute platform (Plattform für die Datenverarbeitung) für Ihre Bereitstellung EC2/On-Premises lautet.

3. Klicken Sie auf der Registerkarte Deployments (Bereitstellungen) auf Create deployment (Bereitstellung erstellen).

   > **Note**
   >
   > Ihre Anwendung muss über eine Bereitstellungsgruppe verfügen, bevor sie bereitgestellt werden kann. Wenn Ihre Anwendung keine Bereitstellungsgruppe hat, wählen Sie auf der Registerkarte Bereitstellungsgruppen die Option Bereitstellungsgruppe erstellen aus. Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy.

4. Wählen Sie unter Deployment group (Bereitstellungsgruppe) eine Bereitstellungsgruppe für diese Bereitstellung aus.
5. Wählen Sie neben Repository type (Repository-Typ) den Repository-Typ aus, in dem Ihre Revision gespeichert ist:

   - Meine Anwendung ist in Amazon S3 gespeichert. Weitere Informationen finden Sie unterGeben Sie Informationen zu einer Revision an, die in einem Amazon S3 S3-Bucket gespeichert ist, und kehren Sie dann zu Schritt 6 zurück.
   - Meine Anwendung ist gespeichert in GitHub — Weitere Informationen finden Sie unterGeben Sie Informationen zu einer Revision an, die in einem GitHub Repository gespeichert ist, und kehren Sie dann zu Schritt 6 zurück.

6. (Optional) Geben Sie im Feld Deployment description (Bereitstellungsbeschreibung) eine Beschreibung für diese Bereitstellung ein.
7. (Optional) Erweitern Sie Bereitstellungskonfiguration überschreiben, um eine Bereitstellungskonfiguration auszuwählen, mit der Sie steuern können, wie der Datenverkehr auf den Amazon EC2- oder lokalen Server verlagert wird, der sich von dem in der Bereitstellungsgruppe angegebenen unterscheidet.

   Weitere Informationen finden Sie unter Arbeiten mit Bereitstellungskonfigurationen in CodeDeploy.
8. a. Wählen Sie „Bereitstellung nicht fehlschlagen, wenn das ApplicationStop Lebenszyklusereignis fehlschlägt“ aus, wenn Sie möchten, dass eine Bereitstellung für eine Instance erfolgreich ist, falls das `ApplicationStop` Lebenszyklusereignis fehlschlägt.

   b. Erweitern Sie die Einstellungen für das Bereitstellungsverhalten, um anzugeben CodeDeploy , wie mit Dateien an einem Zielort für die Bereitstellung umgegangen wird, die nicht Teil der vorherigen erfolgreichen Bereitstellung waren.

      Wählen Sie eine der folgenden Optionen aus:

      - Fehlschlagen der Bereitstellung — Es wird ein Fehler gemeldet und der Bereitstellungsstatus wird auf geändert`Failed`.
      - Inhalt überschreiben — Wenn im Zielverzeichnis eine Datei mit demselben Namen vorhanden ist, wird sie durch die Version aus der Anwendungsrevision ersetzt.
      - Inhalt beibehalten — Wenn im Zielverzeichnis eine Datei mit demselben Namen vorhanden ist, wird die Datei beibehalten und die Version in der Anwendungsrevision wird nicht in die Instanz kopiert.

Weitere Informationen finden Sie unter Rollback-Verhalten bei vorhandenem Inhalt.

9. (Optional) Sie können unter Rollback configuration overrides andere automatische Rollback-Optionen für diese Bereitstellung festlegen, als für die Bereitstellungsgruppe angegeben wurden.

   Hinweise zu Rollbacks in finden Sie CodeDeploy unter Neubereitstellungen und Bereitstellungs-Rollbacks undEine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy.

   Wählen Sie eine der folgenden Optionen aus:

   - Rollback bei fehlgeschlagener Bereitstellung — CodeDeploy stellt die letzte als funktionierend bekannte Version erneut als neue Bereitstellung bereit.
   - Rollback bei Erreichen der Alarmschwellenwerte — Wenn der Bereitstellungsgruppe Alarme hinzugefügt wurden, wird die letzte als CodeDeploy funktionierend bekannte Version bereitgestellt, wenn einer oder mehrere der angegebenen Alarme aktiviert sind.
   - Rollbacks deaktivieren — Führen Sie für diese Bereitstellung keine Rollbacks durch.

10. Klicken Sie auf Start deployment (Bereitstellung starten).

    Informationen zum Verfolgen des Status Ihrer Bereitstellung finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen.

Themen

- Geben Sie Informationen zu einer Revision an, die in einem Amazon S3 S3-Bucket gespeichert ist
- Geben Sie Informationen zu einer Revision an, die in einem GitHub Repository gespeichert ist

Geben Sie Informationen zu einer Revision an, die in einem Amazon S3 S3-Bucket gespeichert ist

Wenn Sie die Schritte unter befolgenErstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (Konsole), gehen Sie wie folgt vor, um Details zu einer Anwendungsversion hinzuzufügen, die in einem Amazon S3 S3-Bucket gespeichert ist.

1. Kopieren Sie den Amazon S3 S3-Link Ihrer Revision in den Revisionsordner. Ermitteln des Link-Wertes:

   a. In einer separaten Browser-Registerkarte:

Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

Navigieren Sie zur Revision und wählen Sie sie aus.

b. Wenn der Bereich Properties nicht sichtbar ist, klicken Sie auf die Schaltfläche Properties.

c. Kopieren Sie im Eigenschaftenbereich den Wert des Link-Felds in das Feld Revisionsort in der CodeDeploy Konsole.

So geben Sie eine ETag-Prüfsumme (eine Dateiprüfsumme) als Teil des Revisionsspeicherorts an:

- Wenn der Wert des Feldes Link auf **`?versionId=`*`versionId`*** endet, fügen Sie **`&etag=`** und den ETag am Ende des Feldwerts für Link hinzu.
- Wenn der Wert im Feld Link keine Versions-ID angibt, fügen Sie **`?etag=`** und den ETag am Ende des Link-Feldwerts hinzu.

> **Note**
>
> Auch wenn es nicht so einfach ist wie das Kopieren des Feldwerts für Link, können Sie auch den Revisionsspeicherort in einem der folgenden Formate eingeben:
> **`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. Wenn eine Nachricht in der Liste File type angezeigt wird, die besagt, dass der Dateityp nicht erkannt werden konnte, wählen Sie den Dateityp der Revision aus. Andernfalls akzeptieren Sie den erkannten Dateityp.

Geben Sie Informationen zu einer Revision an, die in einem GitHub Repository gespeichert ist

Wenn Sie die Schritte unter befolgenErstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (Konsole), gehen Sie wie folgt vor, um Details zu einer in einem GitHub Repository gespeicherten Anwendungsversion hinzuzufügen.

1. Führen Sie unter GitHub Verbinden mit einen der folgenden Schritte aus:

   - Um eine Verbindung für CodeDeploy Anwendungen mit einem GitHub Konto herzustellen, melden Sie sich auf einer anderen Webbrowser-Registerkarte ab GitHub. Geben Sie GitHub unter Konto einen Namen ein, um diese Verbindung zu identifizieren, und wählen Sie dann Connect GitHub. Auf der Webseite werden Sie aufgefordert, die Interaktion mit GitHub Ihrer Anwendung CodeDeploy zu autorisieren. Fahren Sie fort mit Schritt 2.
   - Um eine Verbindung zu verwenden, die Sie bereits erstellt haben, wählen Sie GitHubunter Konto ihren Namen und dann Connect aus GitHub. Fahren Sie fort mit Schritt 4.
   - Um eine Verbindung zu einem anderen GitHub Konto herzustellen, melden Sie sich in einem anderen Webbrowser-Tab ab GitHub. Wählen Sie Connect einem anderen GitHub Konto verbinden und dann Connect GitHub. Fahren Sie fort mit Schritt 2.

2. Wenn Sie aufgefordert werden, sich anzumelden GitHub, folgen Sie den Anweisungen auf der Anmeldeseite. Melden Sie sich mit Ihrem GitHub Benutzernamen oder Ihrer E-Mail-Adresse und Ihrem Passwort an.
3. Wenn eine Authorize application-Seite angezeigt wird, wählen Sie Authorize application.
4. Geben Sie auf der Seite Einrichtung erstellen im Feld Repository-Name den GitHub Benutzer- oder Organisationsnamen ein, der die Revision enthält, gefolgt von einem Schrägstrich (/), gefolgt vom Namen des Repositorys, das die Revision enthält. Falls Sie sich nicht sicher sind, welchen Wert Sie eingeben müssen:

   a. Rufen Sie in einem anderen Webbrowser-Tab Ihr GitHubDashboard auf.

   b. Platzieren Sie unter Your repositories den Mauszeiger auf dem Namen des Ziel-Repositorys. Es erscheint ein Tooltip mit dem Namen des GitHub Benutzers oder der Organisation, gefolgt von einem Schrägstrich (/), gefolgt vom Namen des Repositorys. Geben Sie diesen angezeigten Wert in das Feld Repository name (Repository-Name) ein.

> **Note**
>
> Wenn der Name des Ziel-Repositorys in Ihren Repositorys nicht sichtbar ist, verwenden Sie das GitHubSuchfeld, um den Namen des Ziel-Repositorys und den GitHub Benutzer- oder Organisationsnamen zu finden.

5. Geben Sie unter Commit-ID die ID des Commits ein, das sich auf die Revision im Repository bezieht. Falls Sie sich nicht sicher sind, welchen Wert Sie eingeben müssen:

   a. Rufen Sie in einem anderen Webbrowser-Tab Ihr GitHubDashboard auf.
   b. Wählen Sie unter Your repositories den Namen des Repositorys aus, das den Ziel-Commit enthält.
   c. Suchen und kopieren Sie in der Commit-Liste die Commit-ID, die sich auf die Revision im Repository bezieht. Diese ID ist in der Regel 40 Zeichen lang und besteht aus Buchstaben und Zahlen. (Verwenden Sie nicht die kürzere Version der Commit-ID, die in der Regel die ersten 10 Zeichen der längeren Version der Commit-ID enthält.)
   d. Fügen Sie die Commit-ID in das Feld Commit ID ein.

Erstellen Sie eine Amazon ECS Compute Platform-Bereitstellung (CLI)

Nachdem Sie die Anwendung und die Revision erstellt haben (in Amazon ECS-Bereitstellungen ist dies die AppSpec Datei):

Rufen Sie den Befehl create-deployment auf und geben Sie Folgendes an:

- Einen Anwendungsnamen. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.
- Ein Name einer Bereitstellungsgruppe. Rufen Sie den Befehl list-deployment-groups auf, um eine Liste der Bereitstellungsgruppennamen anzuzeigen.
- Informationen über die bereitzustellende Revision:

  Für in Amazon S3 gespeicherte Revisionen:
  - Der Amazon S3 S3-Bucket-Name, der die Revision enthält.
  - Der Name der hochgeladenen Revision.
  - (Optional) Die Amazon S3 S3-Versions-ID für die Revision. (Wenn die Versions-ID nicht angegeben ist, wird die neueste Version CodeDeploy verwendet.)

- (Optional) Der ETag für die Revision. (Wenn das ETag nicht angegeben wird, führt CodeDeploy keine Objektvalidierung durch.)

  Für Revisionen, die in einer Datei gespeichert sind, die sich nicht in Amazon S3 befindet, benötigen Sie den Dateinamen und den zugehörigen Pfad. Ihre Revisionsdatei wurde mit JSON oder YAML geschrieben, deshalb hat sie wahrscheinlich die Dateinamenerweiterung .json oder .yaml.
- (Optional) Eine Beschreibung für die Bereitstellung.

Die Revisionsdatei kann als in einen Amazon S3 S3-Bucket hochgeladene Datei oder als Zeichenfolge angegeben werden. Die Syntax für die Verwendung als Teil des create-deployment-Befehls lautet:

- Amazon S3 S3-Bucket:

  `version` und `eTag` sind optional.

```
--s3-location bucket=string,key=string,bundleType=JSON|
YAML,version=string,eTag=string
```

- Zeichenfolge:

```
--revision '{"revisionType": "String", "string": {"content":"revision-as-string"}}'
```

> **Note**
>
> Der Befehl create-deployment kann eine Revision aus einer Datei laden. Weitere Informationen finden Sie unter Laden von Parametern aus einer Datei.

Revisionsvorlagen für die AWS Lambda Bereitstellung finden Sie unterFügen Sie eine AppSpec Datei hinzu für AWS Lambda-Bereitstellung. Eine Beispielrevision finden Sie unter AppSpec Dateibeispiel für ein AWS Lambda-Bereitstellung.

Informationen zum Verfolgen des Status Ihrer Bereitstellung finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen.

Erstellen Sie ein AWS Bereitstellung der Lambda Compute Platform (CLI)

Nachdem Sie die Anwendung und die Revision erstellt haben (in AWS Lambda-Bereitstellungen ist dies die AppSpec Datei):

Rufen Sie den Befehl create-deployment auf und geben Sie Folgendes an:

- Einen Anwendungsnamen. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.
- Ein Name einer Bereitstellungsgruppe. Rufen Sie den Befehl list-deployment-groups auf, um eine Liste der Bereitstellungsgruppennamen anzuzeigen.
- Informationen über die bereitzustellende Revision:

  Für in Amazon S3 gespeicherte Revisionen:

  - Der Amazon S3 S3-Bucket-Name, der die Revision enthält.
  - Der Name der hochgeladenen Revision.
  - (Optional) Die Amazon S3 S3-Versions-ID für die Revision. (Wenn die Versions-ID nicht angegeben ist, wird die neueste Version CodeDeploy verwendet.)
  - (Optional) Der ETag für die Revision. (Wenn das ETag nicht angegeben wird, führt CodeDeploy keine Objektvalidierung durch.)

  Für Revisionen, die in einer Datei gespeichert sind, die sich nicht in Amazon S3 befindet, benötigen Sie den Dateinamen und den zugehörigen Pfad. Ihre Revisionsdatei wurde mit JSON oder YAML geschrieben, deshalb hat sie wahrscheinlich die Dateinamenerweiterung .json oder .yaml.
- (Optional) Der Name einer Bereitstellungskonfiguration, die verwendet werden kann. Rufen Sie den Befehl list-deployment-configs auf, um eine Liste der Bereitstellungskonfigurationen anzuzeigen. (Wenn nicht angegeben, wird eine bestimmte CodeDeploy Standardbereitstellungskonfiguration verwendet.)
- (Optional) Eine Beschreibung für die Bereitstellung.

Die Revisionsdatei kann als in einen Amazon S3 S3-Bucket hochgeladene Datei oder als Zeichenfolge angegeben werden. Die Syntax für die Verwendung als Teil des create-deployment-Befehls lautet:

- Amazon S3 S3-Bucket:

  `version` und `eTag` sind optional.

```
--s3-location bucket=string,key=string,bundleType=JSON|
YAML,version=string,eTag=string
```

- Zeichenfolge:

```
--revision '{"revisionType": "String", "string": {"content":"revision-as-string"}}'
```

> **Note**
>
> Der Befehl create-deployment kann eine Revision aus einer Datei laden. Weitere Informationen finden Sie unter Laden von Parametern aus einer Datei.

Revisionsvorlagen für die AWS Lambda Bereitstellung finden Sie unterFügen Sie eine AppSpec Datei hinzu für AWS Lambda-Bereitstellung. Eine Beispielrevision finden Sie unter AppSpec Dateibeispiel für ein AWS Lambda-Bereitstellung.

Informationen zum Verfolgen des Status Ihrer Bereitstellung finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen.

Erstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (CLI)

So verwenden Sie das AWS CLI , um eine Revision auf der EC2/On-Premises Rechenplattform bereitzustellen:

1. Nachdem Sie die Instances vorbereitet, die Anwendung erstellt und die Revision per Push-Verfahren übertragen haben, führen Sie einen der folgenden Schritte aus:
   - Wenn Sie eine Revision aus einem Amazon S3 S3-Bucket bereitstellen möchten, fahren Sie jetzt mit Schritt 2 fort.
   - Wenn Sie eine Revision aus einem GitHub Repository bereitstellen möchten, führen Sie zunächst die Schritte unter aus und fahren Sie dann mit Schritt 2 fort. Eine CodeDeploy Anwendung mit einem GitHub Repository Connect
2. Rufen Sie den Befehl create-deployment auf und geben Sie Folgendes an:
   - `--application-name`: Ein Anwendungsname. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.

- `--deployment-group-name`: Ein Name für eine Amazon EC2 EC2-Bereitstellungsgruppe. Rufen Sie den Befehl list-deployment-groups auf, um eine Liste der Bereitstellungsgruppennamen anzuzeigen.
- `--revision`: Informationen über die Version, die bereitgestellt werden soll:

  Für in Amazon S3 gespeicherte Revisionen:

  - `s3Location`: Der Amazon S3 S3-Bucket-Name, der die Revision enthält.
  - `s3Location --> key`: Der Name der hochgeladenen Revision.
  - `s3Location --> bundleType`: Der Dateityp der hochgeladenen Revision.

    > **Note**
    >
    > Die Dateiformate tar und komprimierte Tar-Archive (.tar und .tar.gz) werden für Windows Server-Instanzen nicht unterstützt.

  - `s3Location --> version`: (Optional) Die Amazon S3 S3-Versions-ID für die Revision. (Wenn die Versions-ID nicht angegeben ist, wird die neueste Version CodeDeploy verwendet.)
  - `s3Location --> eTag`: (Optional) Das ETag für die Revision. (Wenn das ETag nicht angegeben ist, wird die Objektvalidierung CodeDeploy übersprungen.)

  Für Revisionen, die gespeichert sind in: GitHub

  - `gitHubLocation --> repository`: Der GitHub Benutzer- oder Gruppenname, der dem Repository zugewiesen wurde, das die Revision enthält, gefolgt von einem Schrägstrich (/), gefolgt vom Repository-Namen.
  - `gitHubLocation --> commitId`: Die Commit-ID für die Revision.
- `--deployment-config-name`: (Optional) Der Name einer zu verwendenden Bereitstellungskonfiguration. Rufen Sie den Befehl list-deployment-configs auf, um eine Liste der Bereitstellungskonfigurationen anzuzeigen. (Wenn nicht angegeben, wird eine bestimmte CodeDeploy Standardbereitstellungskonfiguration verwendet.)
- `--ignore-application-stop-failures | --no-ignore-application-stop-failures`: (Optional) Gibt an, ob die Bereitstellung auf einer Instance bis zum Ereignis im `BeforeInstall` Bereitstellungslebenszyklus fortgesetzt werden soll, falls das Ereignis im `ApplicationStop` Bereitstellungslebenszyklus fehlschlägt.
- `--description`: (Optional) Eine Beschreibung der Bereitstellung.

- `--file-exists-behavior`: (Optional) Im Rahmen des Bereitstellungsprozesses entfernt der CodeDeploy Agent aus jeder Instanz alle Dateien, die bei der letzten Bereitstellung installiert wurden. Wählen Sie aus, was passiert, wenn Dateien, die nicht Teil einer früheren Bereitstellung waren, an den Zielorten der Bereitstellung angezeigt werden.
- `--target-instances`: Für blue/green Bereitstellungen Informationen über die Instances, die zur Ersatzumgebung in einer blue/green Bereitstellung gehören, einschließlich der Namen einer oder mehrerer Amazon EC2 Auto Scaling Scaling-Gruppen oder des Tag-Filters, der Schlüssel, der Typ und der Wert, die zur Identifizierung von Amazon EC2-Instances verwendet werden.

> **Note**
>
> Verwenden Sie diese Syntax als Teil des create-deployment Aufrufs, um Informationen über eine Revision in Amazon S3 direkt in der Befehlszeile anzugeben. (`version` und `eTag` sind optional.)
>
> ```
> --s3-location bucket=string,key=string,bundleType=tar|tgz|
> zip,version=string,eTag=string
> ```
>
> Verwenden Sie diese Syntax als Teil des create-deployment Aufrufs, um Informationen über eine Revision GitHub direkt in der Befehlszeile anzugeben:
>
> ```
> --github-location repository=string,commitId=string
> ```
>
> Rufen Sie den Befehl list-application-revisions auf, um Informationen über Revisionen zu erhalten, die bereits übertragen wurden.

Informationen zum Verfolgen des Status Ihrer Bereitstellung finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen.

Befehlsreferenz zum Create-Deployment

Die folgende Befehlsstruktur und die folgenden Optionen sind für den `create-deployment` Befehl verfügbar. Weitere Informationen finden Sie in der Referenz „Create-Deployment“ in der AWS CLI Befehlsreferenz.

```
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>]
```

Eine CodeDeploy Anwendung mit einem GitHub Repository Connect

Bevor Sie mithilfe von zum ersten Mal eine Anwendung aus einem GitHub Repository bereitstellen können AWS CLI, müssen Sie zunächst die CodeDeploy Erlaubnis erteilen, im Namen Ihres GitHub Kontos mit GitHub der Anwendung zu interagieren. Dieser Schritt muss für jede Anwendung, die die CodeDeploy Konsole verwendet, einmal abgeschlossen werden.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Wählen Sie Applications (Anwendungen).
3. Wählen Sie unter Anwendungen die Anwendung aus, die Sie mit Ihrem GitHub Benutzerkonto verknüpfen möchten, und wählen Sie Anwendung bereitstellen aus.

   > **Note**
   >
   > Sie erstellen keine Bereitstellung. Dies ist derzeit die einzige Möglichkeit, die CodeDeploy Erlaubnis zur Interaktion GitHub im Namen Ihres GitHub Benutzerkontos zu erteilen.

4. Wählen Sie neben Repository-Typ die Option Meine Anwendungsrevision ist gespeichert in aus GitHub.
5. Wählen Sie Connect GitHub.

> **Note**
>
> Wenn du den Link Mit einem anderen GitHub Konto Connect siehst:
> Möglicherweise haben Sie bereits autorisiert CodeDeploy , GitHub im Namen eines anderen GitHub Kontos mit der Anwendung zu interagieren.
> Möglicherweise haben Sie die Autorisierung für die CodeDeploy Interaktion im Namen des GitHub angemeldeten GitHub Kontos für alle Anwendungen, mit denen Sie verknüpft sind, widerrufen. CodeDeploy
> Weitere Informationen finden Sie unter GitHub Authentifizierung mit Anwendungen in CodeDeploy.

6. Wenn Sie noch nicht angemeldet sind GitHub, folgen Sie den Anweisungen auf der Anmeldeseite.
7. Wählen Sie auf der Seite Authorize application die Option Authorize application.
8. Wenn Sie nun CodeDeploy über die entsprechende Berechtigung verfügen, wählen Sie Abbrechen aus und fahren Sie mit den Schritten unter fortErstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (CLI).

Erstellen Sie eine Amazon blue/green ECS-Bereitstellung über CloudFormation

Sie können es verwenden AWS CloudFormation , um Amazon blue/green ECS-Bereitstellungen über CodeDeploy zu verwalten. Sie generieren Ihre Bereitstellung, indem Sie Ihre grünen und blauen Ressourcen definieren und die Einstellungen für das Routing und die Stabilisierung des Datenverkehrs angeben, die in CloudFormation verwendet werden sollen. In diesem Thema werden die Unterschiede zwischen Amazon blue/green ECS-Bereitstellungen, die von verwaltet werden, CodeDeploy und Bereitstellungen, die von verwaltet werden, behandelt. CloudFormation

Eine schrittweise Anleitung CloudFormation zur Verwaltung Ihrer Amazon blue/green ECS-Bereitstellungen finden Sie unter Automatisieren von blue/green ECS-Bereitstellungen CodeDeploy mithilfe AWS CloudFormation im AWS CloudFormation Benutzerhandbuch.

> **Note**
>
> Die Verwaltung von Amazon blue/green ECS-Bereitstellungen mit CloudFormation ist in der Region Asien-Pazifik (Osaka) nicht verfügbar.

Unterschiede zwischen Amazon blue/green ECS-Bereitstellungen über CodeDeploy und CloudFormation

Die CloudFormation Stack-Vorlage modelliert aufgabenbezogene Ressourcen und Infrastruktur von Amazon ECS sowie die Konfigurationsoptionen für Bereitstellungen. Es gibt also Unterschiede zwischen den standardmäßigen Amazon blue/green ECS-Bereitstellungen und blue/green Bereitstellungen, die dadurch erstellt werden. CloudFormation

Im Gegensatz zu standardmäßigen Amazon blue/green ECS-Bereitstellungen modellieren oder erstellen Sie Folgendes nicht manuell:

- Sie erstellen eine AWS CodeDeploy Anwendung nicht, indem Sie einen Namen angeben, der eindeutig darstellt, was Sie bereitstellen möchten.
- Sie erstellen keine AWS CodeDeploy Bereitstellungsgruppe.
- Sie geben keine Anwendungsspezifikationsdatei (AppSpec Datei) an. Die Informationen, die normalerweise mit der AppSpec Datei verwaltet werden, wie z. B. die gewichteten Konfigurationsoptionen oder Lebenszyklusereignisse, werden vom `AWS::CodeDeploy::BlueGreen` Hook verwaltet.

In dieser Tabelle werden die Unterschiede im Workflow auf hoher Ebene zwischen Bereitstellungstypen zusammengefasst.

Funktion	blue/green Standardbereitstellungen	Blue/green Bereitstellungen durch CloudFormation
Geben Sie den Amazon ECS-Cluster, den Amazon ECS-Service, den Application Load Balancer oder den Network Load Balancer, den Produktio	Erstellen Sie eine CodeDeploy Bereitstellungsgruppe, die diese Ressourcen spezifiziert.	Erstellen Sie eine CloudFormation Vorlage, um diese Ressourcen zu modellieren.

Funktion	blue/green Standardbereitstellungen	Blue/green Bereitstellungen durch CloudFormation
ns-Listener, den Test-Listener und zwei Zielgruppen an.		
Geben Sie die bereitzustellende Änderung an.	Erstellen Sie eine CodeDeploy Anwendung.	Erstellen Sie eine CloudFormation Vorlage, die das Container-Image spezifiziert.
Geben Sie die Amazon ECS-Aufgabendefinition, den Container-Namen und den Container-Port an.	Erstellen Sie eine AppSpec Datei, die diese Ressourcen spezifiziert.	Erstellen Sie eine CloudFormation Vorlage, um diese Ressourcen zu modellieren.
Geben Sie die Optionen für die Verlagerung des Bereitstellungsdatenverkehrs und die Lebenszyklusereignis-Hooks an.	Erstellen Sie eine AppSpec Datei, die diese Optionen spezifiziert.	Erstellen Sie eine CloudFormation Vorlage, die die `AWS::CodeDeploy::BlueGreen` Hook-Parameter verwendet, um diese Optionen anzugeben.
CloudWatch Alarme.	Erstellen Sie einen CloudWatch Alarm, der einen Rollback auslöst.	Konfigurieren Sie einen CloudWatch Alarm auf CloudFormation Stack-Ebene, der ein Rollback auslöst.
Rollback/redeployment.	Legen Sie Optionen für den Rollback und die erneute Bereitstellung fest.	Brechen Sie das Stack-Update in CloudFormation ab.

Überwachung von Amazon blue/green ECS-Bereitstellungen durch CloudFormation

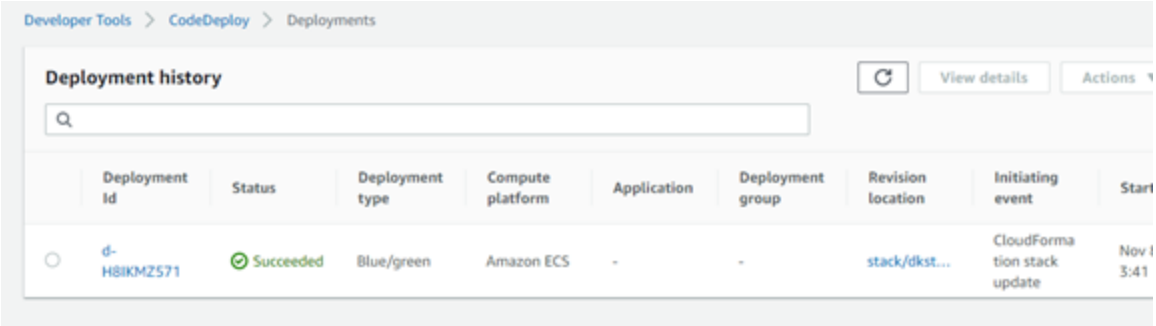
Sie können blue/green Bereitstellungen über CloudFormation und CodeDeploy überwachen. Informationen zur Überwachung durch CloudFormation finden Sie CloudFormation im AWS CloudFormation Benutzerhandbuch unter Überwachen von blue/green Ereignissen.

Um den Bereitstellungsstatus von blue/green Bereitstellungen anzuzeigen, finden Sie in CodeDeploy

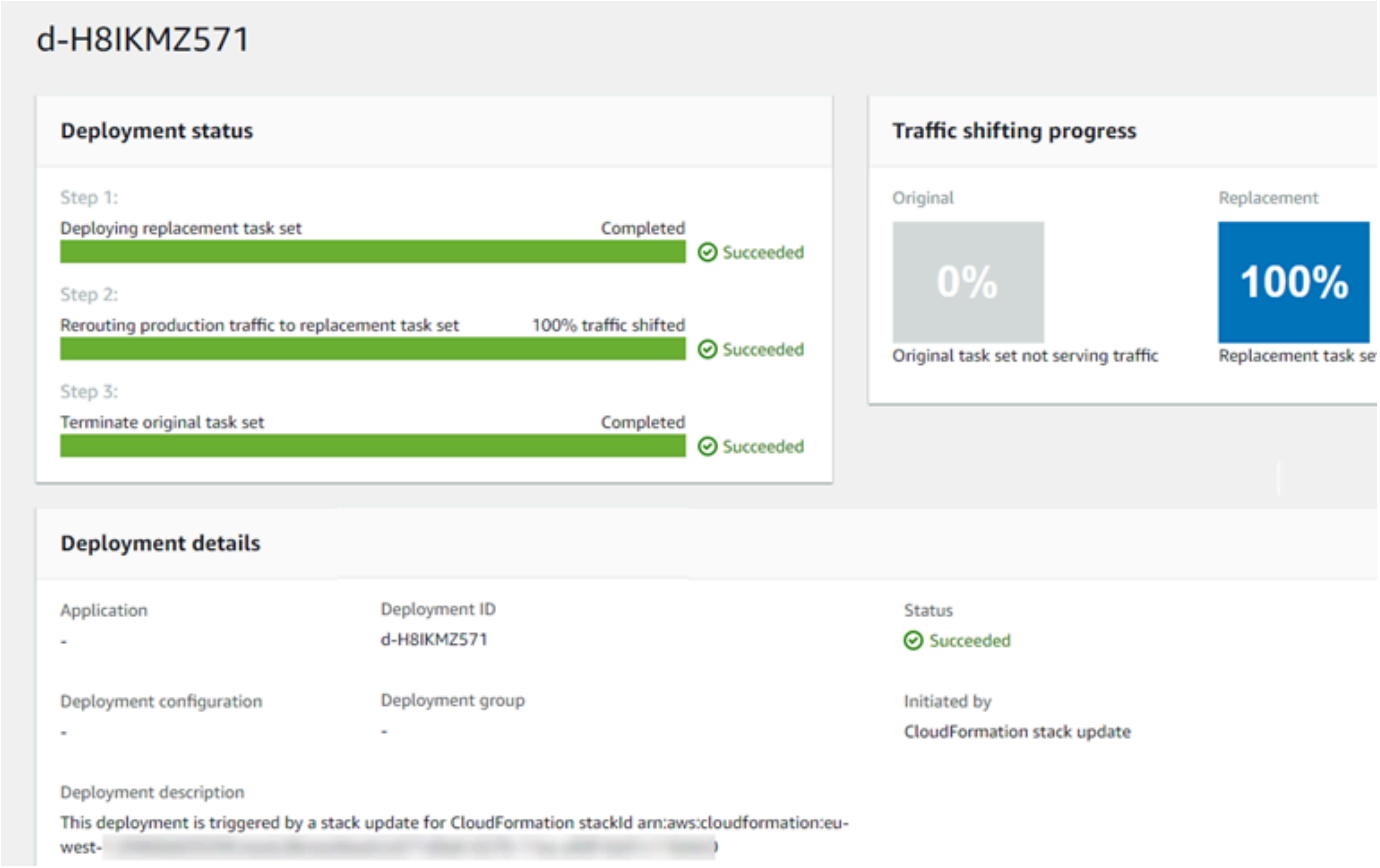
1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Unter Bereitstellungen wird die Bereitstellung angezeigt, die durch das CloudFormation Stack-Update ausgelöst wurde. Wählen Sie die Bereitstellung aus, um den Bereitstellungsverlauf anzuzeigen.



3. Wählen Sie die Bereitstellung aus, um den Status der Datenverkehrsverlagerungen anzuzeigen. Beachten Sie, dass die Anwendung und die Bereitstellungsgruppe nicht erstellt werden.



4. Folgendes gilt für das Zurücksetzen oder Stoppen der Bereitstellung:

   - Die erfolgreiche Bereitstellung wird unter angezeigt CodeDeploy und zeigt, dass die Bereitstellung von CloudFormation initiiert wurde.
   - Wenn Sie die Bereitstellung beenden und rückgängig machen möchten, müssen Sie das Stack-Update in abbrechen CloudFormation.

CodeDeploy Bereitstellungsdetails anzeigen

Sie können die CodeDeploy Konsole, die oder die CodeDeploy APIs verwenden AWS CLI, um Details zu Bereitstellungen anzuzeigen, die mit Ihrem AWS Konto verknüpft sind.

> **Note**
>
> Sie können die EC2/On-Premises Bereitstellungsprotokolle Ihrer Instances an den folgenden Orten einsehen:

- Amazon Linux, RHEL und Ubuntu Server: `/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log`
- Windows Server: C:\ProgramData\ Amazon\ CodeDeploy < DEPLOYMENT-GROUP-ID >< DEPLOYMENT-ID >\ logs\ scripts.log

Weitere Informationen finden Sie unter Analysieren von Protokolldateien zur Untersuchung von Bereitstellungsfehlern auf Instances.

Themen

- Bereitstellungsdetails anzeigen (Konsole)
- Bereitstellungsdetails anzeigen (CLI)

Bereitstellungsdetails anzeigen (Konsole)

So verwenden Sie die CodeDeploy Konsole zum Anzeigen von Bereitstellungsdetails:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich Bereitstellen und wählen Sie dann Bereitstellungen aus.

   > **Note**
   >
   > Wenn keine Einträge angezeigt werden, stellen Sie sicher, dass die richtige Region ausgewählt ist. Wählen Sie in der Navigationsleiste in der Regionsauswahl eine der Regionen aus, die unter Region und Endpoints in der aufgeführt sind. Allgemeine AWS-Referenz CodeDeploy wird nur in diesen Regionen unterstützt.

3. Um weitere Details zu einer einzelnen Bereitstellung anzuzeigen, wählen Sie unter Deployment history (Bereitstellungsverlauf) die Bereitstellungs-ID aus. Sie können auch die Schaltfläche neben der Bereitstellungs-ID auswählen und dann auf View (Anzeigen) klicken.

Bereitstellungsdetails anzeigen (CLI)

Rufen Sie AWS CLI den Befehl oder den `get-deployment` Befehl auf, um die Bereitstellungsdetails anzuzeigen. `batch-get-deployments` Sie können den Befehl `list-deployments` aufrufen, um eine Liste der eindeutigen Bereitstellungs-IDs als Eingabedaten für den Befehl `get-deployment` und den Befehl `batch-get-deployments` zu verwenden.

Um Details zu einer einzelnen Bereitstellung anzuzeigen, rufen Sie den Befehl get-deployment auf und geben Sie dabei die eindeutige Bereitstellungs-ID an. Rufen Sie den Befehl list-deployments auf, um die Bereitstellungs-ID abzurufen.

Um Details zu mehreren Bereitstellungen anzuzeigen, rufen Sie den Befehl batch-get-deployments auf und geben Sie dabei mehrere eindeutige Bereitstellungskennungen an. Rufen Sie den Befehl list-deployments auf, um die Bereitstellungs-IDs abzurufen.

Um eine Liste der Bereitstellungs-IDs anzuzeigen, rufen Sie den Befehl list-deployments auf und geben Sie Folgendes an:

- Der Name der Anwendung, die der Bereitstellung zugeordnet ist. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.
- Der Name der Bereitstellungsgruppe, die der Bereitstellung zugeordnet ist. Rufen Sie den Befehl list-deployment-groups auf, um eine Liste der Bereitstellungsgruppennamen anzuzeigen.
- Optional, ob Sie Details der Bereitstellung nach Bereitstellungstatus einschließen möchten. (Wenn nicht angegeben, werden alle entsprechenden Bereitstellungen aufgelistet, unabhängig von deren Bereitstellungsstatus.)
- Optional, ob Sie Details der Bereitstellung nach deren Erstellungsstartzeit oder Endzeit oder beides einschließen möchten. (Wenn nicht angegeben, werden alle entsprechenden Bereitstellungen aufgelistet, unabhängig von deren Erstellungszeit.)

Protokolldaten für CodeDeploy EC2/On-Premises Bereitstellungen anzeigen

Sie können die durch eine CodeDeploy Bereitstellung erstellten Protokolldaten anzeigen, indem Sie den CloudWatch Amazon-Agenten so einrichten, dass er aggregierte Daten in der CloudWatch Konsole anzeigt, oder indem Sie sich bei einer einzelnen Instance anmelden, um die Protokolldatei zu überprüfen.

> **Note**
>
> Protokolle werden für AWS Lambda- oder Amazon ECS-Bereitstellungen nicht unterstützt. Sie können nur für EC2/On-Premises Bereitstellungen erstellt werden.

Themen

- Protokolldateidaten in der CloudWatch Amazon-Konsole anzeigen
- Logdateien auf einer Instanz anzeigen

Protokolldateidaten in der CloudWatch Amazon-Konsole anzeigen

Wenn der CloudWatch Amazon-Agent auf einer Instance installiert ist, können die Bereitstellungsdaten für alle Bereitstellungen auf dieser Instance in der CloudWatch Konsole angezeigt werden. Der Einfachheit halber empfehlen wir, die Protokolldateien CloudWatch zur zentralen Überwachung zu verwenden, anstatt sie einzeln anzuzeigen. Weitere Informationen finden Sie unter CodeDeploy Agentenprotokolle senden an CloudWatch.

Logdateien auf einer Instanz anzeigen

Um die Bereitstellungsprotokolldaten für eine einzelne Instance zu betrachten, können Sie sich bei der betreffenden Instance anmelden und nach Informationen zu Fehlern und anderen Bereitstellungsereignissen suchen.

Themen

- So zeigen Sie Bereitstellungsprotokolldateien auf Amazon Linux-, RHEL- und Ubuntu-Server-Instances an
- Um Bereitstellungsprotokolldateien auf Windows Server-Instanzen anzuzeigen

So zeigen Sie Bereitstellungsprotokolldateien auf Amazon Linux-, RHEL- und Ubuntu-Server-Instances an

Auf Amazon Linux-, RHEL- und Ubuntu Server-Instances werden Bereitstellungsprotokolle am folgenden Speicherort gespeichert:

`/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log`

Um Bereitstellungsprotokolle auf Amazon Linux-, RHEL- und Ubuntu Server-Instances anzuzeigen oder zu analysieren, melden Sie sich bei der Instance an und geben Sie dann den folgenden Befehl ein, um die CodeDeploy Agenten-Protokolldatei zu öffnen:

```
less /var/log/aws/codedeploy-agent/codedeploy-agent.log
```

Bei Version 2.0.x und höher rotiert das Agent-Protokoll täglich oder wenn die Datei 64 MB erreicht (konfigurierbar). Rotierte Archive erhalten einen Namen `codedeploy-agent.YYYYMMDD.log` (mit einem numerischen Suffix für Rotationen am selben Tag) und werden nach sieben Tagen gelöscht. Das Skriptprotokoll (`scripts.log`) hat eine Größenrotation von 64 MB mit bis zu 8 archivierten Dateien (konfigurierbar).

Geben Sie die folgenden Befehle ein, um in der Protokolldatei nach Fehlermeldungen zu suchen:

Befehl	Ergebnis
& ERROR	Nur die Fehlermeldungen in der Protokoll datei anzeigen. Geben Sie vor und nach der Zeichenfolge **ERROR** ein einzelnes Leerzeichen ein.
/ ERROR	Suchen Sie nach der nächsten Fehlermeldung.[1]
? ERROR	Suchen Sie nach der vorherigen Fehlermel dung.[2] Verwenden Sie ein einzelnes Leerzeich en vor und nach dem Wort. **ERROR**
G	Zum Ende der Protokolldatei navigieren.
g	Zum Anfang der Protokolldatei navigieren.

Befehl	Ergebnis
q	Die Protokolldatei verlassen.
h	Hier erfahren Sie mehr über zusätzliche Befehle.

[1] Geben Sie nach der Eingabe **/ ERROR n** für die nächste Fehlermeldung ein. Geben Sie **N** ein, um nach der vorherigen Fehlermeldung zu suchen.

[2] Geben **? ERROR** Sie nach der Eingabe **n** für die nächste Fehlermeldung oder **N** für die vorherige Fehlermeldung ein.

Sie können auch den folgenden Befehl eingeben, um eine CodeDeploy Skriptprotokolldatei zu öffnen:

```
less /opt/codedeploy-agent/deployment-root/deployment-group-ID/deployment-ID/logs/
scripts.log
```

Geben Sie die folgenden Befehle ein, um in der Protokolldatei nach Fehlermeldungen zu suchen:

Befehl	Ergebnis
&stderr	Nur die Fehlermeldungen in der Protokolldatei anzeigen.
/stderr	Suchen Sie nach der nächsten Fehlermeldung.[1]
?stderr	Suchen Sie nach der vorherigen Fehlermel dung.[2]
G	Zum Ende der Protokolldatei navigieren.
g	Zum Anfang der Protokolldatei navigieren.
q	Die Protokolldatei verlassen.
h	Hier erfahren Sie mehr über zusätzliche Befehle.

Befehl	Ergebnis

[1] Geben Sie nach der Eingabe ein**`/stderr`**, **`n`** um die nächste Fehlermeldung weiterzuleiten. Geben Sie **`N`** ein, um nach der vorherigen Fehlermeldung zu suchen.

[2] Geben **`?stderr`** Sie nach der Eingabe **`n`** für die nächste Fehlermeldung rückwärts ein. Geben Sie **`N`** ein, um nach der vorherigen Fehlermeldung zu suchen.

Um Bereitstellungsprotokolldateien auf Windows Server-Instanzen anzuzeigen

CodeDeploy Agent-Protokolldatei: Auf Windows Server-Instanzen wird die CodeDeploy Agent-Protokolldatei am folgenden Speicherort gespeichert:

Für Version 2.0.x und höher:

`C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log`

Für Version 1.8.x und früher:

`C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt`

Um die CodeDeploy Agent-Protokolldatei auf einer Windows Server-Instanz anzuzeigen oder zu analysieren, melden Sie sich bei der Instanz an, und geben Sie dann den folgenden Befehl ein, um die Datei zu öffnen:

Für Version 2.0.x und höher:

```
notepad C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log
```

Für Version 1.8.x und früher:

```
notepad C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt
```

Um in der Protokolldatei nach Fehlermeldungen zu suchen, drücken Sie STRG+F, geben **`ERROR`** ein und drücken dann die Eingabetaste, um die erste Fehlermeldung zu finden.

CodeDeploy Skript-Protokolldateien: Auf Windows Server-Instanzen werden Bereitstellungsprotokolle am folgenden Speicherort gespeichert:

`C:\ProgramData\Amazon\CodeDeploy\`*`deployment-group-id`*`\`*`deployment-id`*`\logs\scripts.log`

Wobei Folgendes gilt:

- *`deployment-group-id`*ist eine Zeichenfolge wie `examplebf3a9c7a-7c19-4657-8684-b0c68d0cd3c4`
- *`deployment-id`*ist ein Bezeichner wie `d-12EXAMPLE`

Geben Sie den folgenden Befehl ein, um eine CodeDeploy Skriptprotokolldatei zu öffnen:

```
notepad C:\ProgramData\Amazon\CodeDeploy\deployment-group-ID\deployment-ID\logs
\scripts.log
```

Bei Version 2.0.x und höher rotiert das Agent-Protokoll täglich oder wenn die Datei 64 MB erreicht (konfigurierbar). Die Größe des Skriptprotokolls wird auf 64 MB mit bis zu 8 archivierten Dateien rotiert (konfigurierbar).

Um in der Protokolldatei nach Fehlermeldungen zu suchen, drücken Sie STRG+F, geben **`stderr`** ein und drücken dann die Eingabetaste, um die erste Fehlermeldung zu finden.

Stoppen Sie eine Bereitstellung mit CodeDeploy

Sie können die CodeDeploy Konsole, die oder die CodeDeploy APIs verwenden AWS CLI, um Bereitstellungen zu beenden, die mit Ihrem AWS Konto verknüpft sind.

> **Warning**
>
> Das Stoppen einer EC2/On-Premises Bereitstellung kann dazu führen, dass einige oder alle Instances in Ihren Bereitstellungsgruppen in einem unbestimmten Bereitstellungsstatus bleiben. Weitere Informationen finden Sie unter Bereitstellungen wurden gestoppt und sind fehlgeschlagen.

Sie können entweder eine Bereitstellung stoppen oder eine Bereitstellung stoppen und zurücksetzen.

- Stoppen einer Bereitstellung (Konsole)
- Stoppen einer Bereitstellung (CLI)

> **Note**
>
> Wenn es sich bei Ihrer Bereitstellung um eine blue/green Bereitstellung handelt CloudFormation, können Sie diese Aufgabe nicht in der CodeDeploy Konsole ausführen. Gehen Sie zur CloudFormation Konsole, um diese Aufgabe auszuführen.

Stoppen einer Bereitstellung (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Bereitstellungen aus.

   > **Note**
   >
   > Wenn keine Einträge angezeigt werden, stellen Sie sicher, dass die richtige Region ausgewählt ist. Wählen Sie in der Navigationsleiste in der Regionsauswahl eine der Regionen aus, die unter Region und Endpoints in der aufgeführt sind. Allgemeine AWS-Referenz CodeDeploy wird nur in diesen Regionen unterstützt.

3. Wählen Sie die Bereitstellung aus, die Sie stoppen möchten, und gehen Sie wie folgt vor:

   1. Wählen Sie Stop deployment (Bereitstellung stoppen) aus, um die Bereitstellung zu stoppen, ohne ein Rollback durchzuführen.
   2. Wählen Sie Stop and roll back deployment (Bereitstellung mit Rollback stoppen), um die Bereitstellung zu stoppen und ein Rollback durchzuführen.

   Weitere Informationen finden Sie unter Eine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy.

> **Note**
>
> Wenn die Optionen Stop deployment (Bereitstellung stoppen) und Stop and roll back deployment (Bereitstellung mit Rollback stoppen) nicht verfügbar sind, ist die Bereitstellung so weit fortgeschritten, dass sie nicht mehr gestoppt werden kann.

Stoppen einer Bereitstellung (CLI)

Rufen Sie den Befehl stop-deployment auf und geben Sie dabei die Bereitstellungs-ID an. Rufen Sie den Befehl list-deployments auf, um eine Liste der Bereitstellungs-IDs anzuzeigen.

Eine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy

CodeDeploy macht Bereitstellungen rückgängig, indem eine zuvor bereitgestellte Version einer Anwendung erneut als neue Bereitstellung bereitgestellt wird. Diese Rollback-Bereitstellungen sind technisch gesehen neue Bereitstellungen mit neuen Bereitstellungs-IDs und keine wiederhergestellten Versionen einer vorherigen Bereitstellung.

Bereitstellungen können automatisch oder manuell zurückgesetzt werden.

Themen

- Automatische Rollbacks
- Manuelle Rollbacks
- Arbeitsablauf für Rollback und erneutes Deployment
- Rollback-Verhalten bei vorhandenem Inhalt

Automatische Rollbacks

Sie können eine Bereitstellungsgruppe oder Bereitstellung so konfigurieren, dass sie automatisch zurückgesetzt wird, wenn eine Bereitstellung fehlschlägt oder ein definierter Überwachungsschwellenwert erreicht wird. In diesem Fall wird die letzte als funktionierend bekannte Version der Anwendungsrevision bereitgestellt. Sie können automatische Rollbacks konfigurieren, wenn Sie eine Anwendung erstellen und eine Bereitstellungsgruppe erstellen oder aktualisieren.

Bei der Erstellung einer neuen Bereitstellung können Sie den für die Bereitstellungsgruppe konfigurierten automatischen Rollback auch überschreiben.

> **Note**
>
> Sie können Amazon Simple Notification Service verwenden, um eine Benachrichtigung zu erhalten, wenn eine Bereitstellung automatisch zurückgesetzt wird. Weitere Informationen finden Sie unter Monitoring Deployments with Amazon SNS Event Notifications.

Weitere Informationen zur Konfiguration von automatischen Rollbacks finden Sie unter Konfigurieren von erweiterten Optionen für eine Bereitstellungsgruppe.

Manuelle Rollbacks

Wenn Sie keine automatischen Rollbacks festgelegt haben, können Sie eine Bereitstellung manuell zurücksetzen, indem Sie eine neue Bereitstellung mit einer zuvor bereitgestellten Anwendungsrevision erstellen und die Schritte für die erneute Bereitstellung einer Revision befolgen. Dies ist sinnvoll, wenn eine Anwendung in einen unbekannten Status gewechselt hat. Anstatt viel Zeit mit der Fehlerbehebung zu verbringen, können Sie die Anwendung auf einen bekannten Zustand wiederherstellen und erneut bereitstellen. Weitere Informationen finden Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy.

> **Note**
>
> Wenn Sie eine Instanz aus einer Bereitstellungsgruppe entfernen, CodeDeploy wird nichts deinstalliert, was möglicherweise bereits auf dieser Instanz installiert war.

Arbeitsablauf für Rollback und erneutes Deployment

Wenn ein automatisches Rollback initiiert wird oder wenn Sie manuell eine erneute Bereitstellung oder ein manuelles Rollback initiieren, wird CodeDeploy zunächst versucht, alle Dateien, die zuletzt erfolgreich installiert wurden, aus jeder teilnehmenden Instanz zu entfernen. CodeDeploy überprüft dazu die Cleanup-Datei:

`/opt/codedeploy-agent/deployment-root/deployment-instructions/`*`deployment-group-ID`*`-cleanup`Datei (für Amazon Linux-, Ubuntu Server- und RHEL-Instances)

`C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\`*`deployment-group-ID`*`-cleanup`Datei (für Windows Server-Instances)

Falls vorhanden, CodeDeploy verwendet die Cleanup-Datei, um alle aufgelisteten Dateien aus der Instanz zu entfernen, bevor die neue Bereitstellung gestartet wird.

Beispielsweise wurden die ersten beiden Textdateien und zwei Skriptdateien bereits auf einer Amazon EC2 EC2-Instance bereitgestellt, auf der Windows Server ausgeführt wird, und die Skripts erstellten während der Ereignisse im Bereitstellungslebenszyklus zwei weitere Textdateien:

```
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)
```

Die Cleanup-Datei enthält nur die ersten zwei Textdateien und zwei Skriptdateien:

```
c:\temp\a.txt
c:\temp\b.txt
c:\temp\c.bat
c:\temp\d.bat
```

Vor der neuen Bereitstellung CodeDeploy werden nur die ersten beiden Textdateien und die beiden Skriptdateien entfernt, wobei die letzten beiden Textdateien unverändert bleiben:

```
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
```

Im Rahmen dieses Vorgangs CodeDeploy wird nicht versucht, bei nachfolgenden Neubereitstellungen Aktionen rückgängig zu machen oder auf andere Weise miteinander in Einklang zu bringen, unabhängig davon, ob es sich um manuelle oder automatische Rollbacks handelt. Wenn die `d.bat` Dateien `c.bat` und beispielsweise die Logik enthalten, dass die Dateien und, falls sie bereits vorhanden sind, nicht neu erstellt `f.txt` werden, bleiben die alten Versionen von `e.txt`

und bei jeder Ausführung und bei nachfolgenden Bereitstellungen unverändert. `e.txt` `f.txt` CodeDeploy `c.bat` `d.bat` Sie können eine Logik zu `c.bat` und `d.bat` hinzufügen, um vor dem Erstellen neuer Versionen nach alten Versionen von `e.txt` und `f.txt` zu suchen und diese zu löschen.

Rollback-Verhalten bei vorhandenem Inhalt

Im Rahmen des Bereitstellungsprozesses entfernt der CodeDeploy Agent aus jeder Instanz alle Dateien, die bei der letzten Installation installiert wurden. Wenn Dateien, die nicht Teil einer vorherigen Bereitstellung waren, an den Zielorten der Bereitstellung angezeigt werden, können Sie wählen, CodeDeploy was bei der nächsten Bereitstellung mit ihnen geschehen soll:

- Fehlschlagen der Bereitstellung — Es wird ein Fehler gemeldet und der Bereitstellungsstatus wird in Fehlgeschlagen geändert.
- Inhalt überschreiben — Die Version der Datei aus der Anwendungsrevision ersetzt die Version, die sich bereits auf der Instanz befindet.
- Inhalt beibehalten — Die Datei am Zielort wird beibehalten und die Version in der Anwendungsrevision wird nicht in die Instanz kopiert.

Sie können dieses Verhalten wählen, wenn Sie eine Bereitstellung erstellen. Informationen zum Erstellen einer Bereitstellung in der Konsole finden Sie unterErstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (Konsole). Informationen zum Erstellen einer Bereitstellung mit dem AWS CLI finden Sie unterErstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (CLI).

Sie können entscheiden, Dateien für die nächste Bereitstellung beizubehalten, ohne diese zum Anwendungsrevisionspaket hinzufügen zu müssen. Sie können beispielsweise Dateien, die für die Bereitstellung erforderlich sind, aber nicht zum Revisionspaket der Anwendung hinzugefügt wurden, direkt in die Instanz hochladen. Oder Sie können Dateien in die Instanz hochladen, wenn sich Ihre Anwendungen bereits in Ihrer Produktionsumgebung befinden, Sie sie aber CodeDeploy zum ersten Mal verwenden möchten, um sie bereitzustellen.

Im Fall von Rollbacks, bei denen die letzte erfolgreich bereitgestellte Anwendungsrevision aufgrund eines Bereitstellungsfehlers erneut bereitgestellt wird, wird die Inhaltsverarbeitungsoption für diese letzte erfolgreichen Bereitstellung auf die Rollback-Bereitstellung angewandt.

Wenn die Bereitstellung jedoch so konfiguriert war, dass Dateien überschrieben und nicht beibehalten werden, kann während des Rollback ein unerwartetes Ergebnis entstehen. Insbesondere die Dateien, von denen Sie dachten, dass Sie beibehalten würden, könnten von der fehlgeschlagenen

Bereitstellung gelöscht werden. Die Dateien sind nicht in der Instance, wenn die Rollback-Bereitstellung ausgeführt wird.

Im folgenden Beispiel werden drei Bereitstellungen beschrieben. Alle Dateien, die während der zweiten fehlgeschlagenen Bereitstellung überschrieben (gelöscht) werden, sind nicht mehr verfügbar (können nicht aufbewahrt werden), wenn die Anwendungsrevision 1 während der Bereitstellung 3 erneut bereitgestellt wird:

Bereitstellung	Anwendungsrevision.	Option zum Überschreiben von Inhalt	Bereitstellungsstatus	Verhalten und Ergebnis
Bereitstellung 1	Anwendungsrevision 1	AUFBEWAHREN	Erfolgreich	CodeDeploy erkennt Dateien an den Zielspeicherorten, die bei der vorherigen Bereitstellung nicht bereitgestellt wurden. Diese Dateien wurden möglicherweise absichtlich dort platziert, um ein Bestandteil der aktuellen Bereitstellung zu werden. Sie werden als Bestandteil des aktuellen Bereitstellungspakets aufbewahrt und erfasst.

Bereitstellung	Anwendungsrevision.	Option zum Überschreiben von Inhalt	Bereitstellungsstatus	Verhalten und Ergebnis
Bereitstellung 2	Anwendungsrevision 2	ÜBERSCHREIBEN	Fehlgeschlagen	CodeDeploy Löscht während des Bereitstellungsvorgangs alle Dateien, die Teil der vorherigen erfolgreichen Bereitstellung waren. Dies umfasst die Dateien, die während Bereitstellung 1 aufbewahrt wurden. Die Bereitstellung schlägt jedoch aus anderen Gründen fehl.

Bereitstellung	Anwendungsrevision.	Option zum Überschreiben von Inhalt	Bereitstellungsstatus	Verhalten und Ergebnis
Bereitstellung 3	Anwendungsrevision 1	AUFBEWAHREN		Da automatisches Rollback für die Bereitstellung oder Bereitstellungsgruppe aktiviert ist, stellt CodeDeploy die letzte bekannte Anwendungsrevision, Anwendungsrevision 1, bereit. Die Dateien, die Sie in Bereitstellung 1 behalten wollten, wurden jedoch gelöscht, bevor Bereitstellung 2 fehlschlug, und können von AWS CodeDeploy nicht abgerufen werden. Sie können diese Dateien selbst zu der Instance hinzufügen, wenn diese für Anwendungsrevision 1

Bereitstellung	Anwendung srevision.	Option zum Überschreiben von Inhalt	Bereitste llungsstatus	Verhalten und Ergebnis
				benötigt werden, oder Sie können eine neue Anwendung srevision erstellen.

Stellen Sie eine Anwendung in einer anderen Umgebung bereit AWS Konto

Organizations haben in der Regel mehrere AWS Konten, die sie für unterschiedliche Zwecke verwenden (z. B. eines für Systemverwaltungsaufgaben und eines für Entwicklungs-, Test- und Produktionsaufgaben oder eines für Entwicklungs- und Testumgebungen und ein anderes für die Produktionsumgebung).

Sie können zwar ähnliche Arbeiten in verschiedenen Konten ausführen, aber CodeDeploy Bereitstellungsgruppen und die Amazon EC2 EC2-Instances, für die sie bereitgestellt werden, sind strikt an die Konten gebunden, unter denen sie erstellt wurden. Sie können z. B. keine Instance, die Sie in einem Konto gestartet haben, zu einer Bereitstellungsgruppe in einem anderen Konto hinzufügen.

Nehmen wir an, Sie haben zwei AWS Konten: Ihr Entwicklungskonto und Ihr Produktionskonto. Sie arbeiten in erster Linie mit dem Entwicklungskonto, aber Sie wollen in der Lage sein, Bereitstellungen in Ihrem Produktionskonto zu starten, ohne eine umfassende Palette von Anmeldeinformationen und ohne dass Sie sich beim Entwicklungskonto abmelden und beim Produktionskonto wieder anmelden.

Nachdem Sie die Schritte zur kontenübergreifenden Konfiguration ausgeführt haben, können Sie Bereitstellungen initiieren, die zu einem anderen Konto Ihrer Organisation gehören, ohne dass Sie vollständige Anmeldeinformationen für dieses andere Konto benötigen. Sie tun dies teilweise, indem Sie eine von AWS -Security-Token-Service (AWS STS) bereitgestellte Funktion verwenden, die Ihnen temporären Zugriff auf dieses Konto gewährt.

Schritt 1: Erstellen Sie in einem der Konten einen S3-Bucket

Gehen Sie im Entwicklungskonto oder im Produktionskonto wie folgt vor:

- Falls Sie dies noch nicht getan haben, erstellen Sie einen Amazon S3 S3-Bucket, in dem die Anwendungsrevisionen für das Produktionskonto gespeichert werden. Weitere Informationen finden Sie unter Erstellen eines Buckets in Amazon S3. Sie können auch dieselben Buckets und Anwendungsrevisionen für beide Konten verwenden, unter Bereitstellung der gleichen Dateien in der Produktionsumgebung, die Sie in Ihrem Entwicklungskonto getestet und verifiziert haben.

Schritt 2: Erteilen Sie Amazon S3 S3-Bucket-Berechtigungen für das IAM-Instance-Profil des Produktionskontos

Wenn sich der Amazon S3 S3-Bucket, den Sie in Schritt 1 erstellt haben, in Ihrem Produktionskonto befindet, ist dieser Schritt nicht erforderlich. Die Rolle, die Sie später übernehmen, hat bereits Zugriff auf diesen Bucket, da sie sich auch im Produktionskonto befindet.

Wenn Sie den Amazon S3 S3-Bucket im Entwicklungskonto erstellt haben, gehen Sie wie folgt vor:

- Erstellen Sie im Produktionskonto ein IAM-Instance-Profil. Weitere Informationen finden Sie unter Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances.

  > **Note**
  >
  > Notieren Sie sich den ARN für dieses IAM-Instanzprofil. Sie müssen ihn zur Bucket-übergreifenden Richtlinie hinzufügen, die Sie im nächsten Schritt erstellen.

- Geben Sie im Entwicklungskonto dem IAM-Instance-Profil, das Sie gerade in Ihrem Produktionskonto erstellt haben, Zugriff auf den Amazon S3 S3-Bucket, den Sie im Entwicklungskonto erstellt haben. Weitere Informationen finden Sie unter Beispiel 2: Bucket-Besitzer, der kontoübergreifende Bucket-Berechtigungen gewährt.

  Beachten Sie Folgendes, wenn Sie kontenübergreifende Bucket-Berechtigungen erteilen:

  - In der Beispielanleitung stellt Konto A Ihr Entwicklungskonto und Konto B Ihr Produktionskonto dar.
  - Wenn Sie Aufgaben für das Konto A (Entwicklungskonto) durchführen, ändern Sie die folgende Bucket-Richtlinie, um kontenübergreifende Berechtigungen zu erteilen, statt die Musterrichtlinie in der schrittweisen Anleitung zu verwenden.

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 steht für die Kontonummer des Produktionskontos, mit dem Sie gerade das IAM-Instanzprofil erstellt haben.

role-name steht für den Namen des IAM-Instanzprofils, das Sie gerade erstellt haben.

bucket-name steht für den Namen des Buckets, den Sie in Schritt 1 erstellt haben. Stellen Sie sicher, dass Sie `/*` nach dem Namen des Buckets einfügen, um den Zugriff auf alle Dateien im Bucket zu ermöglichen.

Schritt 3: Erstellen Sie Ressourcen und eine kontoübergreifende Rolle im Produktionskonto

In Ihrem Produktionskonto:

- Erstellen Sie Ihre CodeDeploy Ressourcen — Anwendung, Bereitstellungsgruppe, Bereitstellungskonfiguration, Amazon EC2 EC2-Instances, Amazon EC2 EC2-Instance-Profil, Servicerolle usw. — mithilfe der Anweisungen in diesem Handbuch.

- Erstellen Sie eine zusätzliche Rolle, eine kontoübergreifende IAM-Rolle, die ein Benutzer in Ihrem Entwicklungskonto übernehmen kann, um CodeDeploy Vorgänge in diesem Produktionskonto auszuführen.

  Verwenden Sie die Anleitung Exemplarische Vorgehensweise: AWS Kontoübergreifendes Delegieren des Zugriffs mithilfe von IAM-Rollen als Leitfaden für die Erstellung der kontenübergreifenden Rolle. Anstatt die Beispielberechtigungen aus der exemplarischen Vorgehensweise Ihrem Richtliniendokument hinzuzufügen, sollten Sie der Rolle mindestens die folgenden beiden AWS bereitgestellten Richtlinien beifügen:

  - `AmazonS3FullAccess`: Nur erforderlich, wenn der S3-Bucket sich im Entwicklungskonto befindet. Bietet der Rolle des angenommenen Produktionskontos vollen Zugriff auf die Amazon S3 S3-Services und -Ressourcen im Entwicklungskonto, in dem die Revision gespeichert ist.
  - `AWSCodeDeployDeployerAccess`: Ermöglicht es einem Benutzer, Revisionen zu registrieren und bereitzustellen.

  Wenn Sie Bereitstellungsgruppen erstellen und verwalten möchten und nicht nur Bereitstellungen starten möchten, fügen Sie die Richtlinie `AWSCodeDeployFullAccess` anstelle der Richtlinie `AWSCodeDeployDeployerAccess` hinzu. Weitere Informationen zur Verwendung von IAM-verwalteten Richtlinien zur Gewährung von Berechtigungen für CodeDeploy Aufgaben finden Sie unter. AWS verwaltete (vordefinierte) Richtlinien für CodeDeploy

  Sie können zusätzliche Richtlinien zum Durchführen von Aufgaben in anderen AWS -Services mithilfe dieser kontoübergreifenden Rolle hinzufügen.

  Important

  Notieren Sie sich beim Erstellen der kontoübergreifenden IAM-Rolle die Details, die Sie für den Zugriff auf das Produktionskonto benötigen.
  Um die AWS-Managementkonsole zum Rollenwechsel zu verwenden, müssen Sie eine der folgenden Angaben machen:

  - Eine URL für den Zugriff auf das Produktionskonto mit den Anmeldeinformationen der angenommenen Rolle. Sie finden die URL auf der Seite Review (Überprüfen), die am Ende des Erstellungsvorgangs für kontoübergreifende Rollen angezeigt wird.
  - Der Name der kontoübergreifenden Rollen und entweder die Konto-ID oder ein Alias.

  Um die AWS CLI zum Rollenwechsel zu verwenden, müssen Sie Folgendes angeben:

- Den ARN der kontoübergreifenden Rolle, die Sie übernehmen.

Schritt 4: Laden Sie die Anwendungsrevision in den Amazon S3 S3-Bucket hoch

In dem Konto, in dem Sie den Amazon S3 S3-Bucket erstellt haben:

- Laden Sie Ihre Anwendungsrevision in den Amazon S3 S3-Bucket hoch. Weitere Informationen finden Sie unter Eine Revision CodeDeploy auf Amazon S3 übertragen (nur EC2/On-Premises Bereitstellungen).

Schritt 5: Übernehmen Sie die kontoübergreifende Rolle und stellen Sie Anwendungen bereit

Im Entwicklungskonto können Sie das AWS CLI oder das verwenden, um die kontenübergreifende Rolle AWS-Managementkonsole zu übernehmen und die Bereitstellung im Produktionskonto zu initiieren.

Anweisungen zur Verwendung von, um zwischen Rollen AWS-Managementkonsole zu wechseln und Bereitstellungen zu initiieren, finden Sie unter Zu einer Rolle wechseln (AWS-Managementkonsole) und. Erstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (Konsole)

Anweisungen zur Verwendung der, um die kontenübergreifende Rolle AWS CLI zu übernehmen und Bereitstellungen zu initiieren, finden Sie unter Zu einer IAM-Rolle wechseln () und.AWS Command Line InterfaceErstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (CLI)

Weitere Informationen zur Übernahme einer Rolle durch AWS STS finden Sie AssumeRoleim AWS - Security-Token-Service Benutzerhandbuch und assume-role in der Befehlsreferenz.AWS CLI

Verwandte Themen:

- CodeDeploy: Bereitstellung von einem Entwicklungskonto auf ein Produktionskonto

Verwenden Sie den CodeDeploy Agenten, um ein Bereitstellungspaket auf einem lokalen Computer zu validieren

Mit dem CodeDeploy Agenten können Sie Inhalte auf einer Instanz bereitstellen, bei der Sie angemeldet sind. Auf diese Weise können Sie die Integrität einer Anwendungsspezifikationsdatei (AppSpec Datei), die Sie in einer Bereitstellung verwenden möchten, und des Inhalts, den Sie bereitstellen möchten, testen.

Sie müssen keine Anwendung und keine Bereitstellungsgruppe erstellen. Wenn Sie Inhalte bereitstellen möchten, die auf der lokalen Instanz gespeichert sind, benötigen Sie nicht einmal ein AWS Konto. Für die einfachsten Tests können Sie den codedeploy-agent deploy-local Befehl ohne Angabe von Optionen in einem Verzeichnis ausführen, das die AppSpec Datei und den Inhalt enthält, der bereitgestellt werden soll. Das Tool hält Optionen für andere Testfälle bereit.

Eine Überprüfung eines Bereitstellungspakets auf einem lokalen Computer ermöglicht Ihnen Folgendes:

- Testen der Integrität einer Anwendungsrevision
- Testen Sie den Inhalt einer AppSpec Datei.
- Testen Sie es CodeDeploy zum ersten Mal mit Ihrem vorhandenen Anwendungscode.
- Schnelle Bereitstellung von Inhalten, wenn Sie bereits bei einer Instance angemeldet sind

Sie können Deploy-Inhalte verwenden, die auf der lokalen Instance oder in einem unterstützten Remote-Repository-Typ (Amazon S3 S3-Buckets oder öffentliche GitHub Repositorys) gespeichert sind.

Voraussetzungen

Bevor Sie mit einer lokale Bereitstellung beginnen, führen Sie die folgenden Schritte aus:

- Erstellen oder verwenden Sie einen Instance-Typ, der CodeDeploy vom Agenten unterstützt wird. Weitere Informationen finden Sie unter Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy.
- Installieren Sie Version 2.0.0 oder höher des CodeDeploy Agenten. Für den `deploy-local` Unterbefehl ist Version 2.0.0 oder höher erforderlich. Verwenden Sie für Version 1.8.x und früher stattdessen den eigenständigen Befehl. `codedeploy-local` Versionsinformationen finden

Sie unter Versionsverlauf des Agenten CodeDeploy. Installationsinformationen finden Sie unter Installieren Sie den CodeDeploy Agenten.

- Wenn Sie Ihre Inhalte aus einem Amazon S3 S3-Bucket oder GitHub -Repository bereitstellen, stellen Sie einen Benutzer zur Verwendung bereit CodeDeploy. Weitere Informationen finden Sie unter Schritt 1: Einrichtung.
- Wenn Sie Ihre Anwendungsrevision von einem Amazon S3 S3-Bucket aus bereitstellen, erstellen Sie einen Amazon S3 S3-Bucket in der Region, in der Sie arbeiten, und wenden Sie eine Amazon S3 S3-Bucket-Richtlinie auf den Bucket an. Diese Richtlinie gewährt Ihren Instances die erforderlichen Berechtigungen zum Herunterladen der Anwendungsrevision.

  Die folgende Amazon S3 S3-Bucket-Richtlinie ermöglicht beispielsweise, dass jede Amazon EC2 EC2-Instance mit einem angehängten IAM-Instance-Profil, das den ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` enthält, von einer beliebigen Stelle im Amazon S3 S3-Bucket mit dem Namen heruntergeladen werden kann: `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"
                ]
            }
        }
    ]
}
```

  Die folgende Amazon S3 S3-Bucket-Richtlinie ermöglicht es jeder lokalen Instance mit einem zugehörigen IAM-Benutzer, der den ARN `arn:aws:iam::444455556666:user/CodeDeployUser` enthält, von einer beliebigen Stelle im Amazon S3 S3-Bucket mit dem Namen herunterzuladen: `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"
                ]
            }
        }
    ]
}
```

Informationen zum Generieren und Anhängen einer Amazon S3 S3-Bucket-Richtlinie finden Sie unter Beispiele für Bucket-Richtlinien.

- Wenn Sie Ihre Anwendungsrevision aus einem Amazon S3 S3-Bucket oder GitHub -Repository bereitstellen, richten Sie ein IAM-Instance-Profil ein und hängen Sie es an die Instance an. Informationen finden Sie unter Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances, Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy (AWS CLI oder Amazon EC2 EC2-Konsole) und Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy (CloudFormation Vorlage).
- Wenn Sie Ihre Inhalte von bereitstellen GitHub, erstellen Sie ein GitHub Konto und ein öffentliches Repository. Informationen zum Erstellen eines GitHub Kontos finden Sie unter Beitreten GitHub. Informationen zum Erstellen eines GitHub Repositorys finden Sie unter Repo erstellen.

> **Note**
>
> Private Repositorys werden derzeit nicht unterstützt. Wenn Ihr Inhalt in einem privaten GitHub Repository gespeichert ist, können Sie ihn auf die Instanz herunterladen und mit der `--bundle-location` Option den lokalen Pfad angeben.

- Bereiten Sie den Inhalt (einschließlich einer AppSpec Datei) vor, den Sie auf der Instance bereitstellen möchten, und platzieren Sie ihn auf der lokalen Instance, in Ihrem Amazon S3 S3-Bucket oder in Ihrem GitHub Repository. Weitere Informationen finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.

- Wenn Sie andere Werte als die Standardwerte für andere Konfigurationsoptionen verwenden möchten, erstellen Sie die Konfigurationsdatei und platzieren Sie sie auf der Instance (`/etc/codedeploy-agent/conf/codedeployagent.yml`für Amazon Linux-, RHEL- oder Ubuntu-Server-Instances oder `C:\ProgramData\Amazon\CodeDeploy\conf.yml` für Windows Server-Instances). Weitere Informationen finden Sie unter CodeDeploy Referenz zur Agentenkonfiguration.

  > **Note**
  >
  > Wenn Sie eine Konfigurationsdatei auf Amazon Linux-, RHEL- oder Ubuntu-Server-Instances verwenden, müssen Sie entweder:
  >
  > - Verwenden Sie die Variablen `:root_dir:` und `:log_dir:`, um Speicherorte für das Root-Verzeichnis der Bereitstellung und die Protokollverzeichnisordner anzugeben, die von der Standardeinstellung abweichen.
  > - Wird verwendet`sudo`, um CodeDeploy Agentenbefehle auszuführen.

Erstellen Sie eine lokale Bereitstellung

Öffnen Sie auf der Instance, auf der Sie die lokale Bereitstellung erstellen möchten, eine Terminalsitzung (Amazon Linux-, RHEL- oder Ubuntu Server-Instances) oder eine Befehlszeile (Windows Server), um die Tool-Befehle auszuführen.

> **Note**
>
> Der codedeploy-agent deploy-local Befehl verwendet die `codedeploy-agent` Binärdatei an den folgenden Speicherorten:
>
> - Auf Amazon Linux, RHEL oder Ubuntu Server:`/opt/codedeploy-agent/bin/codedeploy-agent`.
> - Auf Windows Server:`C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe`.
>
> Aus Gründen der Abwärtskompatibilität ist der codedeploy-local Befehl auch an den gleichen Stellen verfügbar:

- Auf Amazon Linux, RHEL oder Ubuntu Server befindet sich der Befehl unter`/opt/codedeploy-agent/bin/codedeploy-local`.
- Auf Windows Server befindet sich der Befehl unter`C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-local.exe`.

Die codedeploy-local [options] Syntax funktioniert weiterhin und akzeptiert dieselben Optionen wiecodedeploy-agent deploy-local.

Grundlegende Befehlssyntax

```
codedeploy-agent deploy-local [options]
```

Syntax

```
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>]
```

Optionen

-l, --bundle-location

Gibt den Speicherort des Anwendungsrevisions-Bundles an. Wenn Sie keinen Speicherort angeben, wird vom Tool standardmäßig das Verzeichnis verwendet, in dem Sie derzeit arbeiten. Wenn Sie für `--bundle-location` einen Wert angeben, müssen Sie einen Wert für `--type` angeben.

Beispiele für Bundle-Standortformate:

- Lokale Amazon Linux-, RHEL- oder Ubuntu-Server-Instanz: `/path/to/local/bundle.tgz`
- Lokale Windows Server-Instanz: `C:/path/to/local/bundle`

- Amazon S3 S3-Bucket: `s3://amzn-s3-demo-bucket/bundle.tar`
- GitHub Repository: `https://github.com/`*`account-name`*`/`*`repository-name`*`/`

-t, --type

Das Format des Anwendungsrevisions-Bundles. Unterstützte Typen sind u. a.: `tgz`, `tar`, `zip` und `directory`. Wenn Sie keinen Typ angeben, wird vom Tool standardmäßig `directory` verwendet. Wenn Sie für `--type` einen Wert angeben, müssen Sie einen Wert für `--bundle-location` angeben.

-b, --file-exists-behavior

Dies gibt an, wie Dateien behandelt werden, die bereits an einem Bereitstellungs-Zielspeicherort vorhanden sind, aber nicht Teil einer vorherigen erfolgreichen Bereitstellung waren. Zu den Optionen gehören DISALLOW, OVERWRITE, RETAIN. Weitere Informationen finden Sie in der Datei ExistsBehavior in der AWS CodeDeploy API-Referenz.

-g, --deployment-group

Der Pfad zum Zielordner, in dem der bereitzustellende Inhalt gespeichert wird. Wenn Sie keinen Ordner angeben, wird vom Tool im Root-Verzeichnis Ihrer Bereitstellung ein Ordner mit dem Namen default-local-deployment-group erstellt. Für jede lokale Bereitstellung, die Sie erstellen, erstellt das Tool in diesem Ordner ein Unterverzeichnis mit Namen wie local-1234 (wobei die Nummer die Prozess-ID ist).

-d, --deployment-group-name

Der Name der Bereitstellungsgruppe, der den Lifecycle-Hook-Skripten als Umgebungsvariable zur Verfügung gestellt wird. `DEPLOYMENT_GROUP_NAME` Standardeinstellung: `LocalFleet`.

-a, --anwendungsname

Der Anwendungsname, der den Lifecycle-Hook-Skripten als Umgebungsvariable zur Verfügung gestellt wird. `APPLICATION_NAME` Standardmäßig wird der Speicherort des Pakets verwendet.

-e, --events

Eine Reihe von Override-Lifecycle-Event-Hooks, die Sie der Reihe nach ausführen möchten, anstatt der Ereignisse, die Sie in der AppSpec Datei aufgeführt haben. Mehrere Hooks können durch Kommas getrennt eingegeben werden. Sie können diese Option in folgenden Fällen verwenden:

- Sie möchten einen anderen Satz von Ereignissen ausführen, ohne die AppSpec Datei aktualisieren zu müssen.
- Sie möchten einen einzelnen Event-Hook als Ausnahme von dem ausführen, was in der AppSpec Datei enthalten ist, z. `ApplicationStop` B.

Wenn Sie in der Override-Liste keine Ereignisse angeben DownloadBundleund installieren, werden sie vor allen von Ihnen angegebenen Event-Hooks ausgeführt. Wenn Sie die Optionen DownloadBundleund Install in die Liste der `--events` Optionen aufnehmen, dürfen ihnen nur Ereignisse vorangestellt werden, die normalerweise in CodeDeploy Bereitstellungen vor ihnen ausgeführt werden. Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“.

-c, --agent-configuration-file

Der Speicherort einer Konfigurationsdatei für die Bereitstellung, wenn der Speicherort vom Standardwert abweicht. In einer Konfigurationsdatei werden Alternativen zu anderen Standardwerten und Verhaltensweisen für eine Bereitstellung angegeben.

Standardmäßig werden Konfigurationsdateien in `/etc/codedeploy-agent/conf/codedeployagent.yml` (Amazon Linux-, RHEL- oder Ubuntu-Server-Instances) oder `C:/ProgramData/Amazon/CodeDeploy/conf.yml` (Windows Server) gespeichert. Weitere Informationen finden Sie unter CodeDeploy Referenz zur Agentenkonfiguration.

-A, --appspec-filename

Der Name der Datei. AppSpec Für lokale Bereitstellungen sind die Werte `appspec.yml` und `appspec.yaml` zulässig. Standardmäßig wird die AppSpec Datei aufgerufen`appspec.yml`.

-h, --help

Zeigt eine Zusammenfassung der Hilfeinhalte an.

-v, --version

Zeigt die Versionsnummer des Tools an.

Beispiele

Im Folgenden finden Sie Beispiele für gültige Befehlsformate.

```
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
```

Stellen Sie ein Paket von Amazon S3 bereit:

```
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
```

Stellen Sie ein Paket aus einem öffentlichen GitHub Repository bereit:

```
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
```

Ein Paket unter Angabe mehrerer Lebenszyklusereignisse bereitstellen:

```
codedeploy-agent deploy-local --bundle-location /path/to/local/
bundle.tar --type tar --deployment-group my-deployment-group --events
 DownloadBundle,Install,ApplicationStart,HealthCheck
```

Stoppen Sie eine zuvor bereitgestellte Anwendung mithilfe des ApplicationStop Lebenszyklusereignisses:

```
codedeploy-agent deploy-local --bundle-location /path/to/local/bundle.tgz --type tgz --
deployment-group my-deployment-group --events ApplicationStop
```

Bereitstellung mit eine bestimmten Bereitstellungsgruppen-ID:

```
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
```

Überwachung von Bereitstellungen in CodeDeploy

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer CodeDeploy AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etwaigen Ausfall an mehreren Stellen leichter debuggen können. Bevor Sie mit der Überwachung beginnen CodeDeploy, sollten Sie jedoch einen Überwachungsplan erstellen, der Antworten auf die folgenden Fragen enthält:

- Was sind Ihre Ziele bei der Überwachung?
- Welche Ressourcen werden überwacht?
- Wie oft werden diese Ressourcen überwacht?
- Welche Überwachungstools werden verwendet?
- Wer soll die Überwachungsaufgaben ausführen?
- Wer soll benachrichtigt werden, wenn Fehler auftreten?

Der nächste Schritt besteht darin, eine Ausgangsbasis für die normale CodeDeploy Leistung in Ihrer Umgebung festzulegen, indem Sie die Leistung zu verschiedenen Zeiten und unter verschiedenen Lastbedingungen messen. Speichern Sie bei der Überwachung historische Überwachungsdaten CodeDeploy, damit Sie sie mit aktuellen Leistungsdaten vergleichen, normale Leistungsmuster und Leistungsanomalien identifizieren und Methoden zur Behebung von Problemen entwickeln können.

Wenn Sie beispielsweise verwenden CodeDeploy, können Sie den Status von Bereitstellungen und Zielinstanzen überwachen. Wenn Bereitstellungen oder Instanzen fehlschlagen, müssen Sie möglicherweise eine Anwendungsspezifikationsdatei neu konfigurieren, den CodeDeploy Agenten neu installieren oder aktualisieren, Einstellungen in einer Anwendung oder Bereitstellungsgruppe aktualisieren oder Änderungen an den Instanzeinstellungen oder einer Datei vornehmen. AppSpec

Zur Festlegung eines Grundwertes sollten Sie mindestens die folgenden Elemente überwachen:

- Bereitstellungsereignisse und -status
- Instance-Ereignisse und -status

Automatisierte Überwachungstools

AWS bietet verschiedene Tools, die Sie zur Überwachung verwenden können. CodeDeploy Sie können einige dieser Tools so konfigurieren, dass diese die Überwachung für Sie übernehmen,

während bei anderen Tools ein manuelles Eingreifen nötig ist. Wir empfehlen, dass Sie die Überwachungsaufgaben möglichst automatisieren.

Sie können die folgenden automatisierten Überwachungstools verwenden, um zu beobachten CodeDeploy und zu melden, wenn etwas nicht stimmt:

- Amazon CloudWatch Alarms — Überwachen Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum und führen Sie eine oder mehrere Aktionen aus, die auf dem Wert der Metrik im Verhältnis zu einem bestimmten Schwellenwert über mehrere Zeiträume basieren. Die Aktion ist eine Benachrichtigung, die an ein Amazon Simple Notification Service (Amazon SNS) -Thema oder eine Amazon EC2 Auto Scaling Scaling-Richtlinie gesendet wird. CloudWatch Alarme lösen keine Aktionen aus, nur weil sie sich in einem bestimmten Status befinden. Der Status muss sich geändert haben und für eine bestimmte Anzahl von Zeiträumen beibehalten worden sein. Weitere Informationen finden Sie unter Monitoring Deployments with Amazon CloudWatch Tools.

  Weitere Informationen zur Aktualisierung Ihrer Servicerolle für die CloudWatch -Alarmüberwachung finden Sie unter Gewähren CloudWatch Sie einer CodeDeploy Servicerolle Berechtigungen. Hinweise zum Hinzufügen von CloudWatch Alarmüberwachung zu Ihren CodeDeploy Vorgängen finden Sie unter Erstellen Sie eine Anwendung mit CodeDeployErstellen Sie eine Bereitstellungsgruppe mit CodeDeploy, oderÄndern Sie die Einstellungen der Bereitstellungsgruppe mit CodeDeploy.

- Amazon CloudWatch Logs — Überwachen, speichern und greifen Sie auf Ihre Protokolldateien aus AWS CloudTrail oder anderen Quellen zu. Weitere Informationen finden Sie unter Überwachung von Protokolldateien im CloudWatch Amazon-Benutzerhandbuch.

  Informationen zur Verwendung der CloudWatch Konsole zum Anzeigen von CodeDeploy Protokollen finden Sie unter CodeDeploy CloudWatch Protokolle in der Protokollkonsole anzeigen.

- Amazon CloudWatch Events — Ordnen Sie Ereignisse zu und leiten Sie sie an eine oder mehrere Zielfunktionen oder Streams weiter, um Änderungen vorzunehmen, Statusinformationen zu erfassen und Korrekturmaßnahmen zu ergreifen. Weitere Informationen finden Sie unter Was ist Amazon CloudWatch Events im CloudWatch Amazon-Benutzerhandbuch.

  Informationen zur Verwendung von CloudWatch Events in Ihrem CodeDeploy Betrieb finden Sie unterÜberwachung von Bereitstellungen mit Amazon Events CloudWatch.

- AWS CloudTrail Protokollüberwachung — Teilen Sie Protokolldateien zwischen Konten, überwachen CloudTrail Sie Protokolldateien in Echtzeit, indem Sie sie an CloudWatch Logs senden, schreiben Sie Anwendungen zur Protokollverarbeitung in Java und stellen Sie sicher,

dass sich Ihre Protokolldateien nach der Lieferung von nicht geändert haben CloudTrail. Weitere Informationen finden Sie unter Arbeiten mit CloudTrail Protokolldateien im AWS CloudTrail Benutzerhandbuch.

Informationen zur Verwendung von CloudTrail mit CodeDeploy finden Sie unterMonitoring Deployments.

- Amazon Simple Notification Service — Konfigurieren Sie ereignisgesteuerte Auslöser, um SMS- oder E-Mail-Benachrichtigungen über Bereitstellungs- und Instance-Ereignisse wie Erfolg oder Misserfolg zu erhalten. Weitere Informationen finden Sie unter Thema erstellen und Was ist Amazon Simple Notification Service.

  Informationen zum Einrichten von Amazon SNS SNS-Benachrichtigungen für finden Sie CodeDeploy unterMonitoring Deployments with Amazon SNS Event Notifications.

Manuelle Überwachungstools

Ein weiterer wichtiger Teil der Überwachung CodeDeploy umfasst die manuelle Überwachung der Elemente, die von den CloudWatch Alarmen nicht abgedeckt werden. Die Dashboards CodeDeploy CloudWatch, und andere AWS Konsolen-Dashboards bieten auf einen Blick einen Überblick über den Zustand Ihrer Umgebung. AWS Wir empfehlen, dass Sie auch die Protokolldateien der Bereitstellungen überprüfen. CodeDeploy

- CodeDeploy Die Konsole zeigt:
  - Status von Bereitstellungen
  - Das Datum und die Uhrzeit der zuletzt versuchten und letzten erfolgreichen Bereitstellung einer Revision
  - Die Anzahl der Instances, die in einer Bereitstellung erfolgreich war, fehlgeschlagen ist, übersprungen wurde oder in Bearbeitung ist
  - Der Status lokaler Instances
  - Das Datum und die Uhrzeit, wann lokale Instances registriert oder abgemeldet wurden
- CloudWatch Die Startseite zeigt:
  - Aktuelle Alarme und Status
  - Diagramme mit Alarmen und Ressourcen
  - Servicestatus

Darüber hinaus können CloudWatch Sie Folgendes verwenden:

- Erstellen angepasster Dashboards zur Überwachung der gewünschten Services.
- Aufzeichnen von Metrikdaten, um Probleme zu beheben und Trends zu erkennen
- Suchen und durchsuchen Sie alle Ihre AWS Ressourcenmetriken
- Erstellen und Bearbeiten von Alarmen, um über Probleme benachrichtigt zu werden

Topics

- Monitoring Deployments with Amazon CloudWatch Tools
- Monitoring Deployments
- Monitoring Deployments with Amazon SNS Event Notifications

Überwachung von Bereitstellungen mit Amazon-Tools CloudWatch

Sie können CodeDeploy Bereitstellungen mit den folgenden CloudWatch Tools überwachen: Amazon CloudWatch Events, CloudWatch Alarme und Amazon CloudWatch Logs.

Die Überprüfung der vom CodeDeploy Agenten erstellten Protokolle und der Bereitstellungen kann Ihnen dabei helfen, die Ursachen von Bereitstellungsfehlern zu beheben. Als Alternative zur Überprüfung der CodeDeploy Protokolle für jeweils eine Instanz können Sie CloudWatch Logs verwenden, um alle Protokolle an einem zentralen Ort zu überwachen.

Informationen zur Verwendung von CloudWatch Alarmen und CloudWatch Ereignissen zur Überwachung Ihrer CodeDeploy Bereitstellungen finden Sie in den folgenden Themen.

Themen

- Überwachung von Bereitstellungen mit CloudWatch Alarmen in CodeDeploy
- Überwachung von Bereitstellungen mit Amazon Events CloudWatch

Überwachung von Bereitstellungen mit CloudWatch Alarmen in CodeDeploy

Sie können einen CloudWatch Alarm für eine Instance oder Amazon EC2 Auto Scaling Scaling-Gruppe erstellen, die Sie in Ihren CodeDeploy Vorgängen verwenden. Ein Alarm überwacht eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum und führt eine oder mehrere Aktionen aus, die auf dem Wert der Metrik im Verhältnis zu einem bestimmten Schwellenwert über mehrere

Zeiträume basieren. CloudWatch Alarme rufen Aktionen aus, wenn sich ihr Status ändert (z. B. von `OK` zu`ALARM`).

Mithilfe der systemeigenen CloudWatch Alarmfunktion können Sie alle Aktionen angeben, die unterstützt werden, CloudWatch wenn eine Instance, die Sie in einer Bereitstellung verwenden, ausfällt, z. B. das Senden einer Amazon SNS SNS-Benachrichtigung oder das Stoppen, Beenden, Neustarten oder Wiederherstellen einer Instance. Für Ihren CodeDeploy Betrieb können Sie eine Bereitstellungsgruppe so konfigurieren, dass eine Bereitstellung immer dann gestoppt wird, wenn ein CloudWatch Alarm, den Sie der Bereitstellungsgruppe zuordnen, aktiviert wird.

Sie können einer CodeDeploy Bereitstellungsgruppe bis zu zehn CloudWatch Alarme zuordnen. Wenn einer der angegebenen Alarme aktiviert wird, stoppt die Bereitstellung und der Status wird auf Stopped aktualisiert. Um diese Option verwenden zu können, müssen Sie Ihrer CodeDeploy Servicerolle CloudWatch Berechtigungen erteilen.

Informationen zum Einrichten von CloudWatch Alarmen in der CloudWatch Konsole finden Sie unter CloudWatch Amazon-Alarme erstellen im CloudWatch Amazon-Benutzerhandbuch.

Informationen zum Zuordnen eines CloudWatch Alarms zu einer Bereitstellungsgruppe in CodeDeploy finden Sie unter Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy undÄndern Sie die Einstellungen der Bereitstellungsgruppe mit CodeDeploy.

Themen

- Gewähren CloudWatch Sie einer CodeDeploy Servicerolle Berechtigungen

Gewähren CloudWatch Sie einer CodeDeploy Servicerolle Berechtigungen

Bevor Sie die CloudWatch Alarmüberwachung in Ihren Bereitstellungen verwenden können, muss der Servicerolle, die Sie in Ihrem CodeDeploy Betrieb verwenden, die Berechtigung zum Zugriff auf die CloudWatch Ressourcen erteilt werden.

Um einer Servicerolle CloudWatch Berechtigungen zu erteilen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie in der IAM-Konsole im Navigationsbereich die Option Rollen aus.
3. Wählen Sie den Namen der Servicerolle, die Sie in Ihren AWS CodeDeploy Vorgängen verwenden.

4. Klicken Sie auf der Registerkarte Permissions im Bereich Inline Policies auf Create Role Policy.

   –oder–

   Wenn die Schaltfläche Create Role Policy nicht verfügbar ist, erweitern Sie den Bereich Inline Policies und wählen Sie dann click here aus.
5. Wählen Sie auf der Seite Set Permissions die Option Custom Policy und dann Select aus.
6. Geben Sie auf der Review Policy im Feld Policy Name einen Namen zur Identifizierung dieser Richtlinie ein (z. B. `CWAlarms`).
7. Kopieren Sie das Folgende in das Feld Policy Document:

   JSON

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

8. Klicken Sie auf Apply Policy (Richtlinie anwenden).

Überwachung von Bereitstellungen mit Amazon Events CloudWatch

Sie können Amazon CloudWatch Events verwenden, um Änderungen im Status einer Instance oder einer Bereitstellung (ein „Ereignis“) in Ihrem CodeDeploy Betrieb zu erkennen und darauf zu reagieren. Auf der Grundlage der von Ihnen erstellten Regeln ruft CloudWatch Events dann eine oder mehrere Zielaktionen auf, wenn eine Bereitstellung oder Instance in den Status übergeht, den Sie in einer Regel angeben. Abhängig von der Statusänderung können Sie Benachrichtigungen versenden, Statusinformationen erfassen, Korrekturmaßnahmen ausführen, Ereignisse auslösen oder andere Aktionen ausführen. Sie können die folgenden Zieltypen auswählen, wenn Sie CloudWatch Ereignisse als Teil Ihrer CodeDeploy Operationen verwenden:

- AWS Lambda Funktionen

- Kinesis-Streams
- Amazon-SQS-Warteschlangen
- Built-in Ziele (`EC2 CreateSnapshot API call`,`EC2 RebootInstances API call`,`EC2 StopInstances API call`, und`EC2 TerminateInstances API call`)
- Amazon-SNS-Themen

Nachfolgend finden Sie einige Anwendungsfälle:

- Verwenden Sie eine Lambda-Funktion, um eine Benachrichtigung an einen Slack-Kanal zu senden, wenn eine Bereitstellung fehlschlägt.
- Push-Daten über Bereitstellungen oder Instances in einem Kinesis Stream unterstützen die umfassende Echtzeit-Statusüberwachung.
- Verwenden Sie CloudWatch Alarmaktionen, um Amazon EC2 EC2-Instances automatisch zu stoppen, zu beenden, neu zu starten oder wiederherzustellen, wenn ein von Ihnen festgelegtes Bereitstellungs- oder Instance-Ereignis eintritt.

Im Rest dieses Themas wird das grundlegende Verfahren zum Erstellen einer CloudWatch Ereignisregel für CodeDeploy beschrieben. Bevor Sie jedoch Ereignisregeln zur Verwendung in Ihren CodeDeploy Vorgängen erstellen, sollten Sie Folgendes tun:

- Erfüllen Sie die Voraussetzungen für CloudWatch Ereignisse. Weitere Informationen finden Sie unter Voraussetzungen für Amazon CloudWatch Events.
- Machen Sie sich unter Ereignisse mit Ereignissen, Regeln und Zielen vertraut. CloudWatch Weitere Informationen finden Sie unter Was ist Amazon CloudWatch Events? und Neue CloudWatch Ereignisse — Verfolgen Sie Änderungen an Ihren AWS Ressourcen und reagieren Sie darauf.
- Erstellen Sie das Ziel oder die Ziele für die Ereignisregeln.

So erstellen Sie eine CloudWatch Ereignisregel für CodeDeploy:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Events.
3. Wählen Sie Create rule (Regel erstellen) und dann unter Event selector (Ereignisauswahl) die Option AWS CodeDeploy.
4. Geben Sie einen Detailtyp an:

- Um eine Regel zu erstellen, die für alle Statusänderungen von Instances und Bereitstellungen gilt, wählen Sie Any detail type (Beliebiger Detailtyp), und führen Sie anschließend Schritt 6 aus.
- Um eine Regel zu erstellen, die nur für Instances gilt, wählen Sie Spezifischer Detailtyp und dann CodeDeploy State-changeInstanzbenachrichtigung aus.
- Um eine Regel zu erstellen, die nur für Bereitstellungen gilt, wählen Sie Spezifischer Detailtyp und dann CodeDeploy State-changeBereitstellungsbenachrichtigung aus.

5. Geben Sie die Änderungen an, für die die Regel gilt:

   - Um eine Regel zu erstellen, die für alle Statusänderungen gilt, wählen Sie Any state (Beliebiger Status).
   - Um eine Regel zu erstellen, die nur für einige Statusänderungen gilt, klicken Sie auf Specific state(s) (Beliebige(r) Status), und wählen Sie dann mindestens einen Statuswert aus der Liste aus. Die folgenden Tabelle enthält die wählbaren Statuswerte:

Bereitstellungsstatuswerte	Instance-Statuswerte
FEHLER	FEHLER
START	START
STOP	BEREIT
IN WARTESCHLANGE	ERFOLG
BEREIT	
ERFOLG	

6. Geben Sie an, für welche CodeDeploy Anwendungen die Regel gilt:

   - Um eine Regel zu erstellen, die für alle Anwendungen gilt, wählen Sie Any application (Beliebige Anwendung), und führen Sie anschließend Schritt 8 aus.
   - Um eine Regel zu erstellen, die nur für eine Anwendung gilt, wählen Sie Specific application (Spezifische Anwendung) und dann den Namen der Anwendung aus der Liste.

7. Geben Sie an, für welche Bereitstellungsgruppen die Regel gilt:

- Um eine Regel zu erstellen, die für alle Bereitstellungsgruppen im Zusammenhang mit der ausgewählten Anwendung gilt, wählen Sie Any deployment group (Beliebige Bereitstellungsgruppe).
- Um eine Regel zu erstellen, die nur für eine der Bereitstellungsgruppen im Zusammenhang mit der ausgewählten Anwendung gilt, wählen Sie Specific deployment group(s) (Spezifische Bereitstellungsgruppe(n)) und dann den Namen der Bereitstellungsgruppe aus der Liste.

8. Überprüfen Sie die eingerichteten Regeln, um sicherzustellen, dass sie den Anforderungen Ihrer Ereignisüberwachung entsprechen.
9. Wählen Sie im Bereich Targets die Option Add target*.
10. Wählen Sie in der Liste Select target type (Zieltyp auswählen) den Zieltyp aus, den Sie für die Verwendung mit dieser Regel vorbereitet haben, und konfigurieren Sie dann weitere für diesen Typ erforderliche Optionen.
11. Wählen Sie Details konfigurieren.
12. Geben Sie auf der Seite Regeldetails konfigurieren einen Namen und eine Beschreibung für die Regel ein, und wählen Sie dann das Feld Status aus, um die Regel jetzt zu aktivieren.
13. Wenn Sie mit der Regel zufrieden sind, klicken Sie auf Create rule (Regel erstellen).

Bereitstellungen überwachen mit AWS CloudTrail

CodeDeploy ist CodeDeploy in einen Service integriert CloudTrail, der API-Aufrufe erfasst, die von oder im Namen Ihres AWS Kontos getätigt wurden, und die Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket übermittelt. CloudTrailerfasst API-Aufrufe von der CodeDeploy Konsole, von CodeDeploy Befehlen über die AWS CLI oder direkt von den CodeDeploy APIs aus. Anhand der von gesammelten Informationen können Sie feststellen CloudTrail, an welche Adresse die Anfrage gestellt wurde CodeDeploy, von welcher Quell-IP-Adresse die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde usw. Weitere Informationen darüber CloudTrail, einschließlich der Konfiguration und Aktivierung, finden Sie im AWS CloudTrail Benutzerhandbuch.

CodeDeploy Informationen in CloudTrail

Wenn die CloudTrail Protokollierung in Ihrem AWS Konto aktiviert ist, werden API-Aufrufe von CodeDeploy Aktionen in Protokolldateien nachverfolgt. CodeDeploy Aufzeichnungen werden zusammen mit anderen AWS Serviceaufzeichnungen in einer Protokolldatei geschrieben. CloudTrail bestimmt anhand eines Zeitraums und der Dateigröße, wann eine neue Datei erstellt und in diese geschrieben werden soll.

Alle CodeDeploy Aktionen werden in der AWS CodeDeploy Befehlszeilenreferenz und der AWS CodeDeploy API-Referenz protokolliert und dokumentiert. Beispielsweise generieren Aufrufe zum Erstellen von Bereitstellungen, zum Löschen von Anwendungen und zum Registrieren von Anwendungsrevisionen Einträge in CloudTrail Protokolldateien.

Jeder Protokolleintrag enthält Informationen über den Ersteller der Anforderung. Anhand der Benutzeridentitätsinformationen im Protokoll können Sie feststellen, ob die Anfrage mit Root- oder Benutzeranmeldedaten, mit temporären Sicherheitsanmeldedaten für eine Rolle oder einen Verbundbenutzer oder von einem anderen Dienst gestellt wurde. AWS Weitere Informationen finden Sie im Feld userIdentity in der CloudTrail -Ereignisreferenz.

Sie können Ihre Protokolldateien beliebig lange im Bucket speichern. Sie können aber auch Amazon S3-Lebenszyklusregeln aufstellen, anhand derer die Protokolldateien automatisch archiviert oder gelöscht werden. Standardmäßig wird die serverseitige Verschlüsselung (SSE) von Amazon S3 verwendet, um Ihre Protokolldateien zu verschlüsseln.

Sie können Amazon SNS SNS-Benachrichtigungen CloudTrail veröffentlichen lassen, wenn neue Protokolldateien geliefert werden. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail.

Sie können auch CodeDeploy Protokolldateien aus mehreren AWS Regionen und mehreren AWS Konten in einem einzigen Amazon S3 S3-Bucket zusammenfassen. Weitere Informationen finden Sie unter Empfangen von CloudTrail Protokolldateien aus mehreren Regionen.

Grundlegendes zu CodeDeploy Protokolldateieinträgen

CloudTrail Protokolldateien können einen oder mehrere Protokolleinträge enthalten, wobei jeder Eintrag aus mehreren JSON-formatted Ereignissen besteht. Ein Protokolleintrag stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, etwaige Parameter und das Datum und die Uhrzeit der Aktion. Die Protokolleinträge sind nicht zwangsläufig in einer bestimmten Reihenfolge aufgeführt. Sie sind also kein geordnetes Stacktrace der öffentlichen API-Aufrufe.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die Aktion „Bereitstellungsgruppe CodeDeploy erstellen“ veranschaulicht:

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

Überwachung von Bereitstellungen mit Amazon SNS SNS-Ereignisbenachrichtigungen

Sie können einer CodeDeploy Bereitstellungsgruppe Auslöser hinzufügen, um Benachrichtigungen über Ereignisse im Zusammenhang mit Bereitstellungen oder Instances in dieser Bereitstellungsgruppe zu erhalten. Diese Benachrichtigungen werden an Empfänger gesendet, die ein Amazon SNS SNS-Thema abonniert haben, das Sie in die Aktion des Auslösers aufgenommen haben.

Sie können Benachrichtigungen für CodeDeploy Ereignisse in SMS-Nachrichten oder E-Mail-Nachrichten erhalten. Sie können die JSON-Daten, die beim Eintreten eines bestimmten Ereignisses erstellt werden, auch auf andere Weise verwenden, z. B. beim Senden von Nachrichten an Amazon SQS SQS-Warteschlangen oder beim Aufrufen einer Funktion in. AWS Lambda Für einen Blick auf die Struktur der JSON-Daten für Bereitstellungs- und Instance-Auslöser finden Sie unter JSON-Datenformate für Trigger CodeDeploy.

Sie können einen Auslöser zum Empfangen von Benachrichtigungen in folgenden Fällen verwenden:

- Sie sind ein Entwickler und möchten wissen, wenn eine Bereitstellung fehlschlägt oder angehalten wird, um das Problem zu beheben.
- Sie sind ein Systemadministrator, der wissen muss, wie viele Instances ausfallen, um den Zustand Ihrer Amazon EC2 EC2-Flotte zu überwachen.
- Sie sind Manager und möchten die übersichtliche Anzahl von Bereitstellungs- und Instance-Ereignissen kennen, die Sie durch Filterregeln erhalten, die verschiedene Arten von Benachrichtigungen in Ordner des Desktop-E-Mail-Clients weiterleiten.

Sie können für jede CodeDeploy Bereitstellungsgruppe bis zu 10 Auslöser für jeden der folgenden Ereignistypen erstellen.

Bereitstellungsereignissen	Instance-Ereignisse
• Herzlichen Glückwunsch • Fehler	• Herzlichen Glückwunsch • Fehler

Bereitstellungsereignissen	Instance-Ereignisse
• Gestartet • Angehalten • Rollback • [1] • Alle Bereitstellungsereignissen	• Gestartet • [1] • Alle Instance-Ereignisse

[1] Gilt nur für blue/green Bereitstellungen. Gibt an, dass die neueste Anwendungsrevision auf Instances in einer Ersatzumgebung installiert wurde und der Datenverkehr von der ursprüngl ichen Umgebung jetzt hinter einem Load Balancer umgeleitet werden kann. Weitere Informationen finden Sie unter Arbeiten mit Bereitstellungen in CodeDeploy.

Themen

- Gewähren Sie Amazon SNS SNS-Berechtigungen für eine CodeDeploy Servicerolle
- Erstellen Sie einen Auslöser für ein Ereignis CodeDeploy
- Bearbeiten Sie einen Trigger in einer CodeDeploy Bereitstellungsgruppe
- Löschen Sie einen Trigger aus einer CodeDeploy Bereitstellungsgruppe
- JSON-Datenformate für Trigger CodeDeploy

Gewähren Sie Amazon SNS SNS-Berechtigungen für eine CodeDeploy Servicerolle

Bevor Ihre Auslöser Benachrichtigungen generieren können, muss der Servicerolle, die Sie in Ihren CodeDeploy Abläufen verwenden, die Berechtigung zum Zugriff auf die Amazon SNS SNS-Ressourcen erteilt werden.

So gewähren Sie Amazon SNS SNS-Berechtigungen für eine Servicerolle

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie in der IAM-Konsole im Navigationsbereich die Option Rollen aus.
3. Wählen Sie den Namen der Service-Rolle aus, die Sie in Ihren AWS CodeDeploy -Vorgängen verwenden.

4. Klicken Sie auf der Registerkarte Permissions im Bereich Inline Policies auf Create Role Policy.

   –oder–

   Wenn die Schaltfläche Create Role Policy nicht verfügbar ist, erweitern Sie den Bereich Inline Policies und wählen Sie dann click here aus.
5. Wählen Sie auf der Seite Set Permissions die Option Custom Policy und dann Select aus.
6. Geben Sie auf der Seite Review Policy (Richtlinie überprüfen) im Feld Policy Name (Richtlinienname) einen Namen zur Identifizierung dieser Richtlinie ein (z. B. `SNSPublish`).
7. Kopieren Sie das Folgende in das Feld Policy Document:

   JSON

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

8. Klicken Sie auf Apply Policy (Richtlinie anwenden).

Erstellen Sie einen Auslöser für ein Ereignis CodeDeploy

Sie können einen Auslöser erstellen, der ein Amazon Simple Notification Service (Amazon SNS) -Thema für ein AWS CodeDeploy Bereitstellungs- oder Instance-Ereignis veröffentlicht. Wenn dieses Ereignis eintritt, erhalten alle Abonnenten des zugehörigen Themas Benachrichtigungen über den im Thema angegebenen Endpunkt, z. B. eine SMS- oder E-Mail-Nachricht. Amazon SNS bietet mehrere Methoden zum Abonnieren von Themen.

Bevor Sie einen Trigger erstellen, müssen Sie das Amazon SNS SNS-Thema einrichten, auf das der Trigger verweisen soll. Weitere Informationen finden Sie unter Thema erstellen. Wenn Sie ein Thema erstellen, empfehlen wir Ihnen, ihm einen Namen zu geben, der seinen Zweck kennzeichnet, z. B.

in Formaten wie `Topic-group-us-west-3-deploy-fail` oder`Topic-group-project-2-instance-stop`.

Sie müssen einer CodeDeploy Servicerolle außerdem Amazon SNS SNS-Berechtigungen erteilen, bevor Benachrichtigungen für Ihren Auslöser gesendet werden können. Weitere Informationen finden Sie unter Gewähren Sie Amazon SNS SNS-Berechtigungen für eine CodeDeploy Servicerolle.

Nachdem Sie das Thema erstellt haben, können Sie Abonnenten hinzufügen. Informationen zum Erstellen, Verwalten und Abonnieren von Themen finden Sie unter Was ist Amazon Simple Notification Service.

Erstellen Sie einen Auslöser zum Senden von Benachrichtigungen für CodeDeploy Ereignisse (Konsole)

Sie können die CodeDeploy Konsole verwenden, um Auslöser für ein CodeDeploy Ereignis zu erstellen. Am Ende des Einrichtungsprozesses wird eine Testbenachrichtigung gesendet, um sicherzustellen, dass Berechtigungs- und Auslöserdetails korrekt konfiguriert sind.

Um einen Auslöser für ein CodeDeploy Ereignis zu erstellen

1. Öffnen Sie in der AWS-Managementkonsole die AWS CodeDeploy Konsole.
2. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

3. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
4. Wählen Sie auf der Seite Applications (Anwendungen) den Namen der Anwendung aus, die der Bereitstellungsgruppe zugeordnet ist, der Sie einen Auslöser hinzufügen möchten.
5. Wählen Sie auf der Seite Application details (Anwendungsdetails) die Bereitstellungsgruppe aus, in der Sie einen Auslöser hinzufügen möchten.
6. Wählen Sie Bearbeiten aus.
7. Erweitern Sie Advanced (Erweitert) - optional.

8. Wählen Sie im Bereich Triggers (Auslöser) die Option Create trigger (Auslöser erstellen).
9. Gehen Sie im Bereich Create deployment trigger (Bereitstellungsauslöser erstellen) folgendermaßen vor:

    a. Geben Sie unter Trigger name (Auslösername) einen Namen für den Auslöser ein, an dem seine Funktion leicht zu erkennen ist. Wir empfehlen folgende Formate: `Trigger-group-us-west-3-deploy-fail` oder `Trigger-group-eu-central-instance-stop`.

    b. Wählen Sie unter Ereignisse den Ereignistyp oder die Ereignistypen aus, um das Amazon SNS SNS-Thema zum Senden von Benachrichtigungen auszulösen.

    c. Wählen Sie Amazon SNS SNS-Themen den Namen des Themas aus, das Sie für das Senden von Benachrichtigungen für diesen Auslöser erstellt haben.

    d. Wählen Sie Trigger erstellen. CodeDeploy sendet eine Testbenachrichtigung, um zu bestätigen, dass Sie den Zugriff zwischen CodeDeploy und dem Amazon SNS SNS-Thema korrekt konfiguriert haben. Abhängig vom Endpunkttyp, den Sie für das Thema ausgewählt haben, und ob Sie das Thema abonniert haben, erhalten Sie eine Bestätigung in einer SMS-Nachricht oder einer E-Mail-Nachricht.

10. Wählen Sie Änderungen speichern aus.

Erstellen Sie einen Trigger zum Senden von Benachrichtigungen für CodeDeploy Ereignisse (CLI)

Sie können die CLI verwenden, um beim Erstellen einer Bereitstellungsgruppe Auslöser hinzuzufügen, oder Sie können Auslöser einer bereits vorhandenen Bereitstellungsgruppe hinzufügen.

So erstellen Sie einen Auslöser zum Senden von Benachrichtigungen für eine neue Bereitstellungsgruppe:

Erstellen Sie eine JSON-Datei zum Konfigurieren der Bereitstellungsgruppe, und führen Sie dann den Befehl create-deployment-group mithilfe der Option `--cli-input-json` aus.

Die einfachste Möglichkeit zum Erstellen der JSON-Datei ist es, die Option `--generate-cli-skeleton` zu verwenden, um eine Kopie der JSON-Formats zu erhalten. Stellen Sie dann die erforderlichen Werte in einem Nur-Text-Editor bereit.

1. Führen Sie den folgenden Befehl aus und kopieren Sie die Ergebnisse in einen Nur-Text-Editor.

```
aws deploy create-deployment-group --generate-cli-skeleton
```

2. Fügen Sie der Ausgabe den Namen einer vorhandenen CodeDeploy Anwendung hinzu:

```
{
    "applicationName": "TestApp-us-east-2",
    "deploymentGroupName": "",
    "deploymentConfigName": "",
    "ec2TagFilters": [
        {
            "Key": "",
            "Value": "",
            "Type": ""
        }
    ],
    "onPremisesInstanceTagFilters": [
        {
            "Key": "",
            "Value": "",
            "Type": ""
        }
    ],
    "autoScalingGroups": [
        ""
    ],
    "serviceRoleArn": "",
    "triggerConfigurations": [
        {
            "triggerName": "",
            "triggerTargetArn": "",
            "triggerEvents": [
                ""
            ]
        }
    ]
}
```

3. Geben Sie die Werte für die Parameter an, die Sie konfigurieren möchten.

   Wenn Sie den Befehl create-deployment-group verwenden, müssen Sie mindestens die Werte für die folgenden Parameter angeben:

- `applicationName`: Der Name einer Anwendung, die bereits in Ihrem Konto bereits erstellt wurde.
- `deploymentGroupName`: Ein Name für die Bereitstellungsgruppe, die Sie erstellen.
- `serviceRoleArn`: Der ARN einer vorhandenen Servicerolle, für die CodeDeploy in Ihrem Konto eingerichtet wurde. Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.

Geben Sie im Abschnitt `triggerConfigurations` die Werte für die folgenden Parameter an:

- `triggerName`: Der gewünschte Name für den Auslöser, an dem Sie den Zweck einfach erkennen können. Wir empfehlen folgende Formate: `Trigger-group-us-west-3-deploy-fail` oder `Trigger-group-eu-central-instance-stop`.
- `triggerTargetArn`: Der ARN des Amazon SNS SNS-Themas, das Sie erstellt haben, um es Ihrem Auslöser zuzuordnen, in diesem Format:`arn:aws:sns:us-east-2:444455556666:NewTestTopic`.
- `triggerEvents`: Der Typ von Ereignissen, für die Sie Benachrichtigungen auslösen möchten. Sie können einen oder mehrere Ereignistypen definieren, indem Sie mehrere Ereignistypennamen mit einem Komma voneinander trennen (z. B. `"triggerEvents": ["DeploymentSuccess","DeploymentFailure","InstanceFailure"]`). Wenn Sie mehr als einen Ereignistyp hinzufügen, werden die Benachrichtigungen für alle diese Typen zu dem Thema gesendet, das Sie festgelegt haben. Demnach werden für die einzelnen Ereignistypen keine Benachrichtigungen zu jeweils verschiedenen Themen gesendet. Sie können aus den folgenden Ereignistypen auswählen:
  - DeploymentStart
  - DeploymentSuccess
  - DeploymentFailure
  - DeploymentStop
  - DeploymentRollback
  - DeploymentReady (Gilt nur für Ersatz-Instances in einer blue/green Bereitstellung)
  - InstanceStart
  - InstanceSuccess
  - InstanceFailure
  - InstanceReady (Gilt nur für Ersatzinstanzen in einer blue/green Bereitstellung)

Im folgenden Konfigurationsbeispiel werden die Bereitstellungsgruppe `dep-group-ghi-789-2` für eine Anwendung mit dem Namen `TestApp-us-east-2` sowie ein Auslöser erstellt, der das Senden von Benachrichtigungen auslöst, wenn eine Bereitstellung beginnt, erfolgreich ist oder fehlschlägt:

```
{
    "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. Speichern Sie Ihre Updates als JSON-Datei, und rufen Sie die Datei dann mit der Option `--cli-input-json` auf, wenn Sie den Befehl create-deployment-group ausführen:

> **Important**
>
> Achten Sie darauf, dass `file://` vor dem Dateinamen steht. Dies ist bei diesem Befehl erforderlich.

```
aws deploy create-deployment-group --cli-input-json file://filename.json
```

Am Ende des Erstellungsvorgangs erhalten Sie eine Testbenachrichtigung, um zu überprüfen, ob die Berechtigungs- und Auslöserdetails korrekt konfiguriert sind.

So erstellen Sie einen Auslöser zum Senden von Benachrichtigungen für eine bestehende Bereitstellungsgruppe:

Um Trigger für CodeDeploy Ereignisse AWS CLI zu einer vorhandenen Bereitstellungsgruppe hinzuzufügen, erstellen Sie eine JSON-Datei, um die Bereitstellungsgruppe zu aktualisieren, und führen Sie dann den Befehl update-deployment-group mit der Option aus. `--cli-input-json`

Die einfachste Möglichkeit zum Erstellen der JSON-Datei ist es, den Befehl get-deployment-group auszuführen und so eine Kopie der Bereitstellungsgruppenkonfiguration im JSON-Format zu generieren und dann die Parameterwerte in einem Nur-Text-Editor zu aktualisieren.

1. Führen Sie den folgenden Befehl aus und kopieren Sie die Ergebnisse in einen Nur-Text-Editor.

```
aws deploy get-deployment-group --application-name application --deployment-group-
name deployment-group
```

2. Löschen Sie in der Ausgabe Folgendes:

   - Löschen Sie am Anfang der Ausgabe `{ "deploymentGroupInfo":`.
   - Löschen Sie am Ende der Ausgabe `}`.
   - Löschen Sie die Zeile mit `deploymentGroupId`.
   - Löschen Sie die Zeile mit `deploymentGroupName`.

   Der Inhalt der Textdatei sollte jetzt wie folgt aussehen:

```
{
    "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. Fügen Sie im Abschnitt `triggerConfigurations` Daten für die Parameter `triggerEvents`, `triggerTargetArn` und `triggerName` ein. Hinweise zu den Konfigurationsparametern für Trigger finden Sie unter. TriggerConfig

   Der Inhalt der Textdatei sollte jetzt wie folgt aussehen: Dieser Code löst das Senden von Benachrichtigungen aus, wenn eine Bereitstellung beginnt, erfolgreich ist oder fehlschlägt.

```
{
    "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. Speichern Sie Ihre Updates als JSON-Datei und führen Sie dann den Befehl update-deployment-group mithilfe der Option `--cli-input-json` aus. Stellen Sie sicher, dass Sie die `--current-deployment-group-name` Option angeben und den Namen Ihrer JSON-Datei durch Folgendes ersetzen*`filename`*:

   > **Important**
   >
   > Achten Sie darauf, dass `file://` vor dem Dateinamen steht. Dies ist bei diesem Befehl erforderlich.

   ```
   aws deploy update-deployment-group --current-deployment-group-name deployment-group-name --cli-input-json file://filename.json
   ```

   Am Ende des Erstellungsvorgangs erhalten Sie eine Testbenachrichtigung, um zu überprüfen, ob die Berechtigungs- und Auslöserdetails korrekt konfiguriert sind.

Bearbeiten Sie einen Trigger in einer CodeDeploy Bereitstellungsgruppe

Wenn sich Ihre Benachrichtigungsanforderungen ändern sollten, können Sie die Auslöser ändern, anstatt einen neuen zu erstellen.

Einen CodeDeploy Trigger ändern (CLI)

Um Triggerdetails für CodeDeploy Ereignisse AWS CLI zu ändern, wenn Sie eine Bereitstellungsgruppe aktualisieren, erstellen Sie eine JSON-Datei, um Änderungen an den Eigenschaften der Bereitstellungsgruppe zu definieren, und führen Sie dann den Befehl update-deployment-group mit der Option aus. `--cli-input-json`

Die einfachste Möglichkeit zum Erstellen der JSON-Datei ist es, den Befehl get-deployment-group auszuführen, um die aktuellen Bereitstellungsgruppendetails im JSON-Format erhalten, und anschließend die erforderlichen Werte in einem Nur-Text-Editor zu bearbeiten.

1. Führen Sie den folgenden Befehl aus und ersetzen Sie dabei und durch die Namen Ihrer Anwendung und Bereitstellungsgruppe: *`application deployment-group`*

   ```
   aws deploy get-deployment-group --application-name application --deployment-group-name deployment-group
   ```

2. Kopieren Sie die Ergebnisse des Befehls in einen Nur-Text-Editor und löschen Sie dann Folgendes:

   - Löschen Sie am Anfang der Ausgabe `{ "deploymentGroupInfo":`.
   - Löschen Sie am Ende der Ausgabe `}`.
   - Löschen Sie die Zeile mit `deploymentGroupId`.
   - Löschen Sie die Zeile mit `deploymentGroupName`.

   Der Inhalt der Textdatei sollte jetzt wie folgt aussehen:

```
{
    "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. Ändern Sie nach Bedarf die Parameter. Hinweise zu den Konfigurationsparametern von Triggern finden Sie unter. TriggerConfig

4. Speichern Sie Ihre Updates als JSON-Datei und führen Sie dann den Befehl update-deployment-group mithilfe der Option `--cli-input-json` aus. Stellen Sie sicher, dass Sie die `--current-deployment-group-name` Option angeben und den Namen Ihrer JSON-Datei durch Folgendes ersetzen*`filename`*:

   > **Important**
   >
   > Achten Sie darauf, dass `file://` vor dem Dateinamen steht. Dies ist bei diesem Befehl erforderlich.

   ```
   aws deploy update-deployment-group --current-deployment-group-name deployment-group-name --cli-input-json file://filename.json
   ```

Am Ende des Erstellungsvorgangs erhalten Sie eine Testbenachrichtigung, um zu überprüfen, ob die Berechtigungs- und Auslöserdetails korrekt konfiguriert sind.

Löschen Sie einen Trigger aus einer CodeDeploy Bereitstellungsgruppe

Da pro Bereitstellungsgruppe maximal 10 Auslöser konfiguriert werden können, können Sie Auslöser, die nicht mehr verwendet werden, löschen. Dieser Löschvorgang kann nicht rückgängig gemacht werden, aber Sie können die Auslöser erneut erstellen.

Löschen Sie einen Auslöser aus einer Bereitstellungsgruppe (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Melden Sie sich mit demselben Benutzer an, den Sie eingerichtet habenErste Schritte mit CodeDeploy.

2. Erweitern Sie im Navigationsbereich die Option Bereitstellen und wählen Sie dann Anwendungen aus.
3. Wählen Sie auf der Seite Applications (Anwendungen) den Namen der Anwendung aus, die der Bereitstellungsgruppe zugeordnet ist, in der Sie einen Auslöser löschen möchten.

4. Wählen Sie auf der Seite Application details (Anwendungsdetails) die Bereitstellungsgruppe aus, in der Sie einen Auslöser löschen möchten.
5. Wählen Sie Bearbeiten aus.
6. Erweitern Sie Advanced (Erweitert) - optional.
7. Wählen Sie im Bereich Trigger (Auslöser) den zu löschenden Auslöser und danach Delete trigger (Auslöser löschen) aus.
8. Wählen Sie Änderungen speichern aus.

Löschen eines Triggers aus einer Bereitstellungsgruppe (CLI)

Um die CLI zum Löschen eines Triggers zu verwenden, rufen Sie den Befehl update-deployment-group mit leeren Trigger-Konfigurationsparametern auf und geben Sie an:

- Der Name der Anwendung, die der Bereitstellungsgruppe zugeordnet ist. Rufen Sie den Befehl list-applications auf, um eine Liste von Anwendungsnamen anzuzeigen.
- Der Name der Bereitstellungsgruppe, die der Anwendung zugeordnet ist. Rufen Sie den Befehl list-deployment-groups auf, um eine Liste der Bereitstellungsgruppennamen anzuzeigen.

Beispiel:

```
aws deploy update-deployment-group --application-name application-name --current-
deployment-group-name deployment-group-name --trigger-configurations
```

JSON-Datenformate für Trigger CodeDeploy

Sie können die JSON-Ausgabe verwenden, die erstellt wird, wenn ein Trigger für eine Bereitstellung oder Instance aktiviert wird, in einem benutzerdefinierten Benachrichtigungs-Workflow, z. B. beim Senden von Nachrichten an Amazon SQS SQS-Warteschlangen oder beim Aufrufen einer Funktion in. AWS Lambda

> **Note**
>
> In dieser Anleitung wird nicht behandelt, wie Benachrichtigungen mit JSON konfiguriert werden. Informationen zur Verwendung von Amazon SNS zum Senden von Nachrichten an Amazon SQS SQS-Warteschlangen finden Sie unter Amazon SNS SNS-Nachrichten an Amazon SQS SQS-Warteschlangen senden. Informationen zur Verwendung von Amazon

SNS zum Aufrufen einer Lambda-Funktion finden Sie unter Aufrufen von Lambda-Funktionen mithilfe von Amazon SNS SNS-Benachrichtigungen.

In den folgenden Beispielen wird die Struktur der JSON-Ausgabe bei CodeDeploy-Auslösern dargestellt.

Beispiel Instance-Based für eine JSON-Ausgabe für Trigger

```
{
    "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
    ]
}
```

Beispiel für eine JSON-Ausgabe für Deployment-Based Trigger

```
{
    "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"
    }
}
```

Sicherheit bei AWS CodeDeploy

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Third-party Auditoren testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS Compliance-Programme. Weitere Informationen zu den Compliance-Programmen, die für gelten AWS CodeDeploy, finden Sie unter AWS Services in Umfang nach Compliance-Programmen.
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, die Anforderungen Ihres Unternehmens und die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung anwenden können CodeDeploy. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen CodeDeploy , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer CodeDeploy Ressourcen unterstützen.

Topics

- Datenschutz in AWS CodeDeploy
- Identitäts- und Zugriffsmanagement für AWS CodeDeploy
- Anmeldung und Überwachung CodeDeploy
- Konformitätsüberprüfung für AWS CodeDeploy
- Resilienz in AWS CodeDeploy
- Sicherheit der Infrastruktur in AWS CodeDeploy

Datenschutz in AWS CodeDeploy

Für den Datenschutz in AWS CodeDeploy gilt das Modell der geteilten Verantwortung das von AWS . Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der AWS Cloud alle Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz . Weitere Informationen zum Datenschutz in Europa finden Sie im Zentrum für die Datenschutz-Grundverordnung (DSGVO).

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der API CodeDeploy oder den SDKs arbeiten oder diese anderweitig AWS-Services verwenden. AWS CLI AWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet

werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Richtlinie für den Datenverkehr zwischen Netzwerken

CodeDeploy ist ein vollständig verwalteter Bereitstellungsservice, der EC2-Instances, Lambda-Funktionen, Amazon ECS und lokale Server unterstützt. Bei EC2-Instances und lokalen Servern kommuniziert ein hostbasierter Agent über TLS mit. CodeDeploy

Derzeit ist für die Kommunikation zwischen dem Agenten und dem Service eine ausgehende Internetverbindung erforderlich, sodass der Agent mit den öffentlichen Endpunkten CodeDeploy und den Amazon S3-Serviceendpunkten kommunizieren kann. In einer Virtual Private Cloud kann dies mit einem Internet-Gateway, einer Site-to-Site VPN-Verbindung zu Ihrem Unternehmensnetzwerk oder einer direkten Verbindung erfolgen.

Der CodeDeploy Agent unterstützt HTTP-Proxys.

Amazon VPC-Endpunkte, betrieben von AWS PrivateLink, sind CodeDeploy in bestimmten Regionen verfügbar. Details hierzu finden Sie unter Verwendung CodeDeploy mit Amazon Virtual Private Cloud.

> **Note**
>
> Der CodeDeploy Agent ist nur erforderlich, wenn Sie ihn auf einer EC2/On-premises Amazon-Rechenplattform bereitstellen. Der Agent ist für Bereitstellungen, die Amazon ECS oder die AWS Lambda Rechenplattform verwenden, nicht erforderlich.

Verschlüsselung im Ruhezustand

Der Kundencode ist nicht gespeichert in CodeDeploy. Als Bereitstellungsdienst sendet er Befehle an den CodeDeploy Agenten, CodeDeploy der auf EC2-Instances oder lokalen Servern ausgeführt wird. Der CodeDeploy Agent führt die Befehle dann mithilfe von TLS aus. Servicemodelldaten für Bereitstellungen, Bereitstellungskonfigurationen, Bereitstellungsgruppen, Anwendungen und Anwendungsrevisionen werden in Amazon DynamoDB gespeichert und im Ruhezustand unter Verwendung von AWS-eigener Schlüssel, Eigentümer und Verwaltet von verschlüsselt. CodeDeploy Weitere Informationen finden Sie unter s.AWS-eigener Schlüssel

Verschlüsselung während der Übertragung

Der CodeDeploy Agent initiiert die gesamte Kommunikation CodeDeploy über Port 443. Der Agent fragt CodeDeploy ab und wartet auf einen Befehl. Der CodeDeploy Agent ist Open Source. Die gesamte Kommunikation zwischen Service und Service sowie zwischen Client und Service wird während der Übertragung mit TLS verschlüsselt. Dies schützt Kundendaten bei der Übertragung zwischen CodeDeploy und anderen Diensten wie Amazon S3.

Verwaltung von Verschlüsselungsschlüsseln

Sie müssen keine Verschlüsselungsschlüssel verwalten. Die Daten des CodeDeploy Servicemodells werden mit einem verschlüsselt AWS-eigener Schlüssel, der Eigentümer ist und von diesem verwaltet wird CodeDeploy. Weitere Informationen finden Sie unter AWS-eigener Schlüssel s.

Identitäts- und Zugriffsmanagement für AWS CodeDeploy

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. CodeDeploy IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- Wie AWS CodeDeploy funktioniert mit IAM
- AWS verwaltete (vordefinierte) Richtlinien für CodeDeploy
- CodeDeploy Aktualisierungen zu AWS Verwaltete Richtlinien
- AWS CodeDeploy Beispiele für identitätsbasierte Richtlinien
- Fehlerbehebung AWS CodeDeploy Identität und Zugriff
- CodeDeploy Referenz zu Berechtigungen
- Cross-service verwirrter Abgeordneter, Prävention

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe Fehlerbehebung AWS CodeDeploy Identität und Zugriff).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe Wie AWS CodeDeploy funktioniert mit IAM).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe AWS CodeDeploy Beispiele für identitätsbasierte Richtlinien).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Benutzer und Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-

Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer für den Zugriff AWS mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identity-based Richtlinien

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität (Benutzer, Gruppe oder Rolle) anhängen. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identity-based Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mehreren Identitäten zugeordnet sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.
- Service-Kontrollrichtlinien (SCPs) – SCPs legen die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in AWS Organizations fest. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Ressourcen-Kontrollrichtlinien (RCPs) – RCPs definieren die maximale Anzahl an Berechtigungen, die Ressourcen in Ihren Konten zur Verfügung stehen. Weitere Informationen finden Sie unter Ressourcen-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere Arten von Richtlinien für eine Anfrage gelten, sind die sich daraus ergebenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter Bewertungslogik für Richtlinien im IAM-Benutzerhandbuch.

Wie AWS CodeDeploy funktioniert mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf zu verwalten CodeDeploy, sollten Sie wissen, mit welchen IAM-Funktionen Sie verwenden können. CodeDeploy Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Themen

- CodeDeploy identitätsbasierte Richtlinien
- CodeDeploy ressourcenbasierte Richtlinien
- Autorisierung auf der Grundlage von Tags CodeDeploy
- CodeDeploy IAM-Rollen

CodeDeploy identitätsbasierte Richtlinien

Mit identitätsbasierten IAM-Richtlinien können Sie zulässige oder verweigerte Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zugelassen oder verweigert werden. CodeDeploy unterstützt Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu den Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der Referenz zu den IAM-JSON-Richtlinienelementen im IAM-Benutzerhandbuch.

Aktionen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Bei Richtlinienaktionen wird das `codedeploy:` Präfix vor der Aktion CodeDeploy verwendet. Die `codedeploy:GetApplication`-Berechtigung erteilt dem Benutzer zum Beispiel Berechtigungen zum Ausführen der `GetApplication`-Operation. Richtlinienerklärungen müssen `Action` entweder ein `NotAction` Oder-Element enthalten. CodeDeploy definiert eigene Aktionen, die Aufgaben beschreiben, die Sie mit diesem Dienst ausführen können.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommata:

```
"Action": [
      "codedeploy:action1",
      "codedeploy:action2"
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie die folgende Aktion zur Angabe aller Aktionen einschließen, die mit dem Wort `Describe` beginnen:

```
"Action": "ec2:Describe*"
```

Eine Liste der CodeDeploy Aktionen finden Sie AWS CodeDeploy im IAM-Benutzerhandbuch unter Definierte Aktionen von.

Eine Tabelle, in der alle CodeDeploy API-Aktionen und die Ressourcen, für die sie gelten, aufgeführt sind, finden Sie unterCodeDeploy Referenz zu Berechtigungen.

Ressourcen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Sie können in Ihrer Anweisung beispielsweise eine Bereitstellungsgruppe (*myDeploymentGroup*) angeben, indem Sie ihren ARN wie folgt verwenden:

```
"Resource": "arn:aws:codedeploy:us-
west-2:123456789012:deploymentgroup:myApplication/myDeploymentGroup"
```

Sie können auch alle Bereitstellungsgruppen angeben, die zu einem Konto gehören, indem Sie das Platzhalterzeichen (*) wie folgt verwenden:

```
"Resource": "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:*"
```

Wenn Sie alle Ressourcen angeben möchten oder wenn eine API-Aktion keine ARNs unterstützt, verwenden Sie das Platzhalterzeichen (*) im `Resource`-Element, wie folgt:

```
"Resource": "*"
```

Einige CodeDeploy API-Aktionen akzeptieren mehrere Ressourcen (z. B.`BatchGetDeploymentGroups`). Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie die ARNs mit Komma, wie folgt:

```
"Resource": ["arn1", "arn2"]
```

CodeDeploy bietet eine Reihe von Operationen für die Arbeit mit den CodeDeploy Ressourcen. Eine Liste der verfügbaren Operationen finden Sie unter CodeDeploy Referenz zu Berechtigungen.

Eine Liste der CodeDeploy Ressourcentypen und ihrer ARNs finden Sie unter Resources Defined by AWS CodeDeploy im IAM-Benutzerhandbuch. Informationen zu den Aktionen, in denen Sie den ARN jeder Ressource angeben können, finden Sie unter Aktionen Definiert von AWS CodeDeploy.

CodeDeploy Ressourcen und Operationen

CodeDeployIn ist die primäre Ressource eine Bereitstellungsgruppe. In einer Richtlinie verwenden Sie einen Amazon-Ressourcennamen (ARN), um die Ressource zu identifizieren, für die die Richtlinie gilt. CodeDeploy unterstützt andere Ressourcen, die mit Bereitstellungsgruppen verwendet werden können, einschließlich Anwendungen, Bereitstellungskonfigurationen und Instances. Diese werden als Unterressourcen bezeichnet. Diesen Ressourcen und Subressourcen sind eindeutige ARNs zugeordnet. Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) in der Allgemeine Amazon Web Services-Referenz.

Ressourcentyp	ARN-Format
Bereitstellungsgruppe	`arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`
Anwendung	`arn:aws:codedeploy:region:account-id:application:application-name`

Ressourcentyp	ARN-Format
Bereitstellungskonfiguration	`arn:aws:codedeploy:region:account-id:deploymentconfig:deployment-configuration-name`
Instance	`arn:aws:codedeploy:region:account-id:instance/instance-ID`
Alle Ressourcen CodeDeploy	`arn:aws:codedeploy:*`
Alle CodeDeploy Ressourcen, die dem angegebenen Konto in der angegebenen Region gehören	`arn:aws:codedeploy:region:account-id:*`

> **Note**
>
> Die meisten Dienste in AWS behandeln einen Doppelpunkt (:) oder einen Schrägstrich (/) als dasselbe Zeichen in ARNs. CodeDeploy Verwendet jedoch eine exakte Übereinstimmung in den Ressourcenmustern und Regeln. Verwenden Sie also die richtigen ARN-Zeichen zum Erstellen von Ereignismustern, sodass sie mit der ARN-Syntax in der Ressource übereinstimmen.

Bedingungsschlüssel

CodeDeploy stellt keine dienstspezifischen Bedingungsschlüssel bereit, unterstützt aber die Verwendung einiger globaler Bedingungsschlüssel. Weitere Informationen finden Sie unter Globale AWS -Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

Beispiele

Beispiele für CodeDeploy identitätsbasierte Richtlinien finden Sie unter. AWS CodeDeploy Beispiele für identitätsbasierte Richtlinien

CodeDeploy ressourcenbasierte Richtlinien

CodeDeploy unterstützt keine ressourcenbasierten Richtlinien. Ein Beispiel für eine detaillierte Seite mit ressourcenbasierten Richtlinien finden Sie unter Verwenden ressourcenbasierter Richtlinien für. AWS Lambda

Autorisierung auf der Grundlage von Tags CodeDeploy

CodeDeploy unterstützt nicht das Markieren von Ressourcen oder das Steuern des Zugriffs auf der Grundlage von Tags.

CodeDeploy IAM-Rollen

Eine IAM-Rolle ist eine Entität in Ihrem AWS Konto, die über bestimmte Berechtigungen verfügt.

Verwenden temporärer Anmeldeinformationen mit CodeDeploy

Sie können temporäre Anmeldeinformationen verwenden, um sich über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende Rolle anzunehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie AWS STS API-Operationen wie AssumeRoleoder aufrufen GetFederationToken.

CodeDeploy unterstützt die Verwendung temporärer Anmeldeinformationen.

Service-linked Rollen

CodeDeploy unterstützt keine dienstbezogenen Rollen.

Servicerollen

Dieses Feature ermöglicht einem Service das Annehmen einer Servicerolle in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Servicerollen werden in Ihrem AWS Konto angezeigt und gehören dem Konto. Das bedeutet, dass ein Benutzer die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktionalität des Dienstes beeinträchtigen.

CodeDeploy unterstützt Servicerollen.

Auswahl einer IAM-Rolle in CodeDeploy

Wenn Sie eine Bereitstellungsgruppenressource in erstellen CodeDeploy, müssen Sie eine Rolle auswählen, mit der Sie in Ihrem Namen auf Amazon EC2 zugreifen können CodeDeploy . Wenn Sie

zuvor eine Servicerolle oder serviceverknüpfte Rolle erstellt haben, stellt Ihnen CodeDeploy eine Liste mit Rollen bereit, aus denen Sie wählen können. Es ist wichtig, eine Rolle zu wählen, die Zugriff zum Starten und Stoppen von EC2-Instances ermöglicht.

AWS verwaltete (vordefinierte) Richtlinien für CodeDeploy

AWS adressiert viele gängige Anwendungsfälle durch die Bereitstellung eigenständiger IAM-Richtlinien, die von erstellt und verwaltet AWS werden. Diese AWS verwalteten Richtlinien gewähren Berechtigungen für allgemeine Anwendungsfälle, sodass Sie nicht erst untersuchen müssen, welche Berechtigungen erforderlich sind. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

Themen

- Liste von AWS verwaltete Richtlinien für CodeDeploy
- CodeDeploy verwaltete Richtlinien und Benachrichtigungen

Liste von AWS verwaltete Richtlinien für CodeDeploy

Die folgenden AWS verwalteten Richtlinien, die Sie Benutzern in Ihrem Konto zuordnen können, sind spezifisch für CodeDeploy:

- `AWSCodeDeployFullAccess`: Gewährt uneingeschränkten Zugriff auf CodeDeploy.

> **Note**
>
> AWSCodeDeployFullAccess gewährt keine Berechtigungen für Operationen in anderen Services, die für die Bereitstellung Ihrer Anwendungen erforderlich sind, wie Amazon EC2 und Amazon S3, sondern nur für spezifische Operationen von. CodeDeploy

- `AWSCodeDeployDeployerAccess`: Erteilt die Genehmigung zur Registrierung und Bereitstellung von Revisionen.

- `AWSCodeDeployReadOnlyAccess`: Gewährt schreibgeschützten Zugriff auf CodeDeploy.

- `AWSCodeDeployRole`: Erlaubt CodeDeploy :

- lesen Sie die Tags auf Ihren Instances oder identifizieren Sie Ihre Amazon EC2-Instances anhand der Amazon EC2 Auto Scaling-Gruppennamen
- Amazon EC2 Auto Scaling-Gruppen, Lifecycle-Hooks, Skalierungsrichtlinien und Warm-Pool-Funktionen lesen, erstellen, aktualisieren und löschen
- Informationen zu Amazon SNS SNS-Themen veröffentlichen
- Informationen über CloudWatch Amazon-Alarme abrufen
- Ressourcen im Elastic Load Balancing Service lesen und aktualisieren

Die Richtlinie enthält den folgenden Code:

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`: CodeDeploy Erteilt Zugriffsberechtigungen AWS Lambda und alle anderen Ressourcen, die für eine Bereitstellung erforderlich sind.

- `AWSCodeDeployRoleForECS`: CodeDeploy Erteilt die Genehmigung für den Zugriff auf Amazon ECS und alle anderen Ressourcen, die für eine Bereitstellung erforderlich sind.

- `AWSCodeDeployRoleForECSLimited`: CodeDeploy Erteilt die Genehmigung für den Zugriff auf Amazon ECS und alle anderen Ressourcen, die für eine Bereitstellung erforderlich sind, mit den folgenden Ausnahmen:
  - In dem `hooks` Abschnitt der AppSpec Datei `CodeDeployHook_` können nur Lambda-Funktionen verwendet werden, deren Namen mit beginnen. Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“ für eine Amazon ECS-Bereitstellung.

  - S3-Bucket-Zugriff ist auf S3-Buckets mit dem Registrierungs-Tag `UseWithCodeDeploy` beschränkt, für das der Wert `true` festgelegt ist. Weitere Informationen finden Sie unter Objekt-Tagging.

- `AmazonEC2RoleforAWSCodeDeployLimited`: CodeDeploy Erteilt die Erlaubnis, Objekte in einem CodeDeploy Amazon S3 S3-Bucket abzurufen und aufzulisten. Die Richtlinie enthält den folgenden Code:

  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"
                }
            }
        }
    ]
}
```

Berechtigungen für einige Aspekte des Bereitstellungsprozesses werden zwei anderen Rollentypen erteilt, die im Namen von agieren CodeDeploy:

- Ein IAM-Instance-Profil ist eine IAM-Rolle, die Sie Ihren Amazon EC2 EC2-Instances zuordnen. Dieses Profil beinhaltet die erforderlichen Berechtigungen für den Zugriff auf die Amazon S3 S3-Buckets oder GitHub Repositorys, in denen die Anwendungen gespeichert sind. Weitere Informationen finden Sie unter Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances.
- Eine Servicerolle ist eine IAM-Rolle, die einem AWS Service Berechtigungen erteilt, sodass dieser auf Ressourcen zugreifen kann. AWS Die Richtlinien, die Sie der Servicerolle zuordnen, bestimmen, auf welche AWS Ressourcen der Dienst zugreifen kann und welche Aktionen er mit diesen Ressourcen ausführen kann. Denn CodeDeploy eine Servicerolle wird für Folgendes verwendet:
  - Um entweder die auf die Instances angewendeten Tags oder die Amazon EC2 Auto Scaling Scaling-Gruppennamen zu lesen, die den Instances zugeordnet sind. Auf diese Weise können CodeDeploy Instances identifiziert werden, auf denen Anwendungen bereitgestellt werden können.
  - Um Operationen auf Instances, Amazon EC2 Auto Scaling Scaling-Gruppen und Elastic Load Balancing Balancing-Load Balancers durchzuführen.
  - Um Informationen zu Amazon SNS SNS-Themen zu veröffentlichen, sodass Benachrichtigungen gesendet werden können, wenn bestimmte Bereitstellungs- oder Instance-Ereignisse eintreten.
  - Zum Abrufen von Informationen über CloudWatch Alarme, um die Alarmüberwachung für Bereitstellungen einzurichten.

  Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.

Sie können auch benutzerdefinierte IAM-Richtlinien erstellen, um Berechtigungen für CodeDeploy Aktionen und Ressourcen zu erteilen. Sie fügen diese benutzerdefinierten Richtlinien den IAM-Rollen hinzu und weisen die Rollen dann Benutzern oder Gruppen zu, die die Berechtigungen benötigen.

CodeDeploy verwaltete Richtlinien und Benachrichtigungen

CodeDeploy unterstützt Benachrichtigungen, um Benutzer über wichtige Änderungen an Bereitstellungen zu informieren. Zu den verwalteten Richtlinien CodeDeploy gehören auch Richtlinienerklärungen für die Benachrichtigungsfunktion. Weitere Informationen finden Sie unter Was sind Benachrichtigungen?.

Berechtigungen für Benachrichtigungen in verwalteten Richtlinien mit vollem Zugriff

Die von `AWSCodeDeployFullAccess` verwaltete Richtlinie enthält die folgenden Anweisungen, um den vollständigen Zugriff auf Benachrichtigungen zu ermöglichen. Benutzer, für die diese verwaltete Richtlinie gilt, können auch Amazon SNS SNS-Themen für Benachrichtigungen erstellen und verwalten, Themen für Benutzer abonnieren und abbestellen und Themen auflisten, die als Ziele für Benachrichtigungsregeln ausgewählt werden können.

```
    {
        "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": "*"
    }
```

Berechtigungen für Benachrichtigungen in schreibgeschützten verwalteten Richtlinien

Die verwaltete Richtlinie `AWSCodeDeployReadOnlyAccess` enthält die folgenden Anweisungen, um schreibgeschützten Zugriff auf Benachrichtigungen zu ermöglichen. Benutzer mit dieser verwalteten Richtlinie können Benachrichtigungen für Ressourcen anzeigen, sie können sie jedoch nicht erstellen, verwalten oder abonnieren.

```
   {
        "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": "*"
    }
```

Berechtigungen für Benachrichtigungen in anderen verwalteten Richtlinien

Die `AWSCodeDeployDeployerAccess` verwaltete Richtlinie umfasst die folgenden Anweisungen, die es Benutzern ermöglichen, Benachrichtigungen für Ressourcen zu erstellen, zu aktualisieren, zu abonnieren und anzuzeigen, diese jedoch nicht zu löschen. Benutzer, für die diese verwaltete Richtlinie gilt, können auch Amazon SNS SNS-Themen für Benachrichtigungen erstellen und verwalten.

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `codestar-notifications:CreateNotificationRule`— Ermöglicht Prinzipalen das Erstellen von Benachrichtigungen.
- `codestar-notifications:DescribeNotificationRule`— Ermöglicht Prinzipalen das Abrufen von Informationen über Benachrichtigungen.
- `codestar-notifications:UpdateNotificationRule`— Ermöglicht Prinzipalen, Benachrichtigungen zu aktualisieren.
- `codestar-notifications:Subscribe`— Ermöglicht Prinzipalen, Benachrichtigungsaktualisierungen zu abonnieren.
- `codestar-notifications:Unsubscribe`— Ermöglicht Prinzipalen das Abbestellen von Benachrichtigungsaktualisierungen.
- `codestar-notifications:ListNotificationRules`— Ermöglicht Prinzipalen das Abrufen der Liste der Benachrichtigungsregeln.
- `codestar-notifications:ListTargets`— Ermöglicht Prinzipalen das Abrufen der Zielliste.
- `codestar-notifications:ListTagsforResource`— Ermöglicht es den Prinzipalen, die Liste der Tags abzurufen.
- `codestar-notifications:ListEventTypes`— Ermöglicht es den Prinzipalen, die Liste der Ereignistypen abzurufen.
- `chatbot:DescribeSlackChannelConfiguration`— Ermöglicht Prinzipalen das Abrufen von Informationen über Slack-Channel-Konfigurationen.
- `sns:ListTopics`— Ermöglicht Principals, die Liste der Amazon SNS SNS-Themen für Benachrichtigungen abzurufen.

```
  {
     "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" : "*"
    }
```

Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für CodeStar AWS-Benachrichtigungen.

CodeDeploy Aktualisierungen zu AWS Verwaltete Richtlinien

Zeigt Details zu Aktualisierungen der AWS verwalteten Richtlinien an, die CodeDeploy seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst vorgenommen wurden. Um automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf CodeDeploy Dokumentverlauf.

Änderungen	Beschreibung	Date
`AWSCodeDeployDeployerAccess` verwaltete Richtlinie — Aktualisierungen vorhandener Richtlinien	Die `codestar-notifications:NotificationsForResource` Aktion wurde aktualisiert, um Änderungen der IAM-Richtlinienvalidierung zu unterstützen. Die ursprüngliche Ressource `arn:aws:codedeploy:*` wurde auf `arn:aws:codedeploy:*:*:application:*` aktualisiert. Weitere Informationen zu dieser Richtlinie finden Sie unter Berechtigungen für Benachrichtigungen in anderen verwalteten Richtlinien.	16. Dezember 2024
`AWSCodeDeployReadOnlyAccess` verwaltete	Die `codestar-notifications:NotificationsForResource` Aktion	16. Dezember 2024

Änderungen	Beschreibung	Date
Richtlinie — Aktualisierungen der bestehenden Richtlinie	wurde aktualisiert, um Änderungen der IAM-Richtlinienvalidierung zu unterstützen. Die ursprüngliche Ressource `arn:aws:codedeploy:*` wurde auf `arn:aws:codedeploy:*:*:application:*` aktualisiert. Weitere Informationen zu dieser Richtlinie finden Sie unter Berechtigungen für Benachrichtigungen in schreibgeschützten verwalteten Richtlinien.	

Änderungen	Beschreibung	Date
`AWSCodeDeployFullAccess` verwaltete Richtlinie — Aktualisierungen der bestehenden Richtlinie	Die `codestar-notifications:NotificationsForResource` Aktion wurde aktualisiert, um Änderungen der IAM-Richtlinienvalidierung zu unterstützen. Die ursprüngliche Ressource `arn:aws:codedeploy:*` wurde auf `arn:aws:codedeploy:*:*:application:*` aktualisiert. Weitere Informationen zu dieser Richtlinie finden Sie unter Berechtigungen für Benachrichtigungen in verwalteten Richtlinien mit vollem Zugriff.	16. Dezember 2024
`AWSCodeDeployRole` verwaltete Richtlinie — Aktualisierungen der bestehenden Richtlinie	Der Richtlinienerklärung wurden die `elasticloadbalancing:DescribeTargetGroupAttributes` Aktionen `elasticloadbalancing:DescribeLoadBalancerAttributes` und hinzugefügt, um Elastic Load Balancing Balancing-Änderungen zu unterstützen. Weitere Informationen zu dieser Richtlinie finden Sie unter AWSCodeDeployRole.	16. August 2023

Änderungen	Beschreibung	Date
`AWSCodeDeployFullAccess` verwaltete Richtlinie — Aktualisierungen der bestehenden Richtlinie	Die `chatbot:ListMicrosoftTeamsChannelConfigurations` Aktion wurde zur Richtlinienerklärung hinzugefügt, um Benachrichtigungsänderungen zu unterstützen. Weitere Informationen zu dieser Richtlinie finden Sie unter AWSCodeDeployRole.	11. Mai 2023
`AWSCodeDeployRole` verwaltete Richtlinie — Aktualisierungen der bestehenden Richtlinie	Die `autoscaling:CreateOrUpdateTags` Aktion wurde zur Richtlinienerklärung hinzugefügt, um Amazon EC2 Auto Scaling Scaling-Autorisierungsänderungen zu unterstützen. Weitere Informationen zu dieser Richtlinie finden Sie unter AWSCodeDeployRole.	3. Februar 2023
`AmazonEC2RoleforAWSCodeDeployLimited` verwaltete Richtlinie — Aktualisierungen der bestehenden Richtlinie	Die `s3:ListBucket` Aktion, die die `s3:ExistingObjectTag/UseWithCodeDeploy` Bedingung enthält, wurde aus der Richtlinienerklärung entfernt. Weitere Informationen zu dieser Richtlinie finden Sie unter AmazonEC2RoleforAWSCodeDeployLimited.	22. November 2021

Änderungen	Beschreibung	Date
`AWSCodeDeployRole` verwaltete Richtlinie — Aktualisierungen der vorhandenen Richtlinie	Die `autoscaling:PutWarmPool` Aktion zur Unterstützung des Hinzufügens von Warm-Pools zu Amazon EC2 Auto Scaling Scaling-Gruppen für blue/green Bereitstellungen wurde hinzugefügt. Unnötige doppelte Aktionen wurden entfernt.	18. Mai 2021
CodeDeploy hat begonnen, Änderungen zu verfolgen	CodeDeploy hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	18. Mai 2021

AWS CodeDeploy Beispiele für identitätsbasierte Richtlinien

Standardmäßig sind Benutzer nicht berechtigt, CodeDeploy Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mit der AWS-Managementkonsole AWS CLI, oder AWS API ausführen. Sie müssen IAM-Richtlinien erstellen, die IAM-Rollen die Erlaubnis gewähren, API-Operationen auf den angegebenen Ressourcen auszuführen, die sie benötigen. Anschließend müssen Sie diese IAM-Rollen Benutzern oder Gruppen zuordnen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von Richtlinien auf der JSON-Registerkarte im IAM-Benutzerhandbuch.

In werden identitätsbasierte Richtlinien verwendet CodeDeploy, um die Berechtigungen für die verschiedenen Ressourcen im Zusammenhang mit dem Bereitstellungsprozess zu verwalten. Sie können den Zugriff auf die folgenden Ressourcentypen steuern:

- Anwendungen und Anwendungsrevisionen
- Bereitstellungen

- Bereitstellungskonfigurationen
- Instances und lokale Instances

Die von Ressourcenrichtlinien gesteuerten Funktionen variieren je nach Ressourcentyp, wie in der folgenden Tabelle dargestellt:

Ressourcentypen	Capabilities
Alle	Anzeigen und Auflisten von Details zu Ressourcen
Anwendungen Bereitstellungskonfigurationen Bereitstellungsgruppen	Erstellen von -Ressourcen Löschen von Ressourcen
Bereitstellungen	Erstellen von Bereitstellungen Stoppen von Bereitstellungen
Anwendungsrevisionen	Registrieren von Anwendungsrevisionen
Anwendungen Bereitstellungsgruppen	Aktualisieren von Ressourcen
On-premises Instanzen	Hinzufügen von Tags zu Instances Entfernen von Tags zu Instances Registrieren von Instances Abmelden von Instances

Das folgende Beispiel zeigt eine Berechtigungsrichtlinie, die es einem Benutzer gestattet, die Bereitstellungsgruppe mit dem Namen **`WordPress_DepGroup`** zu löschen, die der Anwendung **`WordPress_App`** in der Region **`us-west-2`** zugeordnet ist.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:DeleteDeploymentGroup"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:deploymentgroup:WordPress_App/
WordPress_DepGroup"
      ]
    }
  ]
}
```

Themen

- Customer-managed Beispiele für politische Maßnahmen
- Best Practices für Richtlinien
- Verwenden der Konsole CodeDeploy
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Customer-managed Beispiele für politische Maßnahmen

In diesem Abschnitt finden Sie Beispielrichtlinien, die Berechtigungen für verschiedene CodeDeploy Aktionen gewähren. Diese Richtlinien funktionieren, wenn Sie die CodeDeploy API, AWS SDKs oder die AWS CLI verwenden. Sie müssen zusätzliche Berechtigungen für Aktionen erteilen, die Sie in der Konsole ausführen. Weitere Informationen zum Erteilen von Konsolenberechtigungen finden Sie unter Verwenden der Konsole CodeDeploy.

> **Note**
>
> In allen Beispielen werden die Region USA West (Oregon) (`us-west-2`) und fiktive Konto-IDs verwendet.

Beispiele

- Beispiel 1: Erlauben Sie die Erlaubnis, CodeDeploy Operationen in einer einzelnen Region auszuführen
- Beispiel 2: Erlauben Sie die Erlaubnis, Revisionen für eine einzelne Anwendung zu registrieren
- Beispiel 3: Erlauben Sie die Erlaubnis, Bereitstellungen für eine einzelne Bereitstellungsgruppe zu erstellen

Beispiel 1: Erlauben Sie die Erlaubnis, CodeDeploy Operationen in einer einzelnen Region auszuführen

Das folgende Beispiel gewährt Berechtigungen zur Durchführung von CodeDeploy Vorgängen nur in der **`us-west-2`** Region:

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:*"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:*"
      ]
    }
  ]
}
```

Beispiel 2: Erlauben Sie die Erlaubnis, Revisionen für eine einzelne Anwendung zu registrieren

Das folgende Beispiel erteilt Berechtigungen zum Registrieren von Anwendungsrevisionen für alle Anwendungen, die mit **`Test`** in der Region **`us-west-2`** beginnen:

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:RegisterApplicationRevision"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:application:Test*"
      ]
    }
  ]
}
```

Beispiel 3: Erlauben Sie die Erlaubnis, Bereitstellungen für eine einzelne Bereitstellungsgruppe zu erstellen

Das folgende Beispiel gewährt die Berechtigung zum Erstellen von Bereitstellungen für die angegebene Bereitstellungsgruppe, die mit der genannten Anwendung **`WordPress_DepGroup`** verknüpft ist**`WordPress_App`**, für die angegebene benutzerdefinierte Bereitstellungskonfiguration und für alle Anwendungsversionen**`ThreeQuartersHealthy`**, die mit der genannten Anwendung verknüpft sind. **`WordPress_App`** Alle diese Ressourcen befinden sich in der Region **`us-west-2`**.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:CreateDeployment"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:deploymentgroup:WordPress_App/
WordPress_DepGroup"
      ]
```

```
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:GetDeploymentConfig"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-
west-2:111122223333:deploymentconfig:ThreeQuartersHealthy"
      ]
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:GetApplicationRevision"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:application:WordPress_App"
      ]
    }
  ]
}
```

Best Practices für Richtlinien

Identity-based Richtlinien legen fest, ob jemand CodeDeploy Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer

Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.

- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Konsole CodeDeploy

Wenn Sie die CodeDeploy Konsole verwenden, müssen Sie über Mindestberechtigungen verfügen, die es Ihnen ermöglichen, andere AWS Ressourcen für Ihr AWS Konto zu beschreiben. Für die Verwendung CodeDeploy in der CodeDeploy Konsole benötigen Sie Berechtigungen für die folgenden Dienste:

- Amazon EC2 Auto Scaling

- AWS CodeDeploy
- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- AWS Identity and Access Management
- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon CloudWatch

Wenn Sie eine IAM-Richtlinie erstellen, die restriktiver ist als die erforderlichen Mindestberechtigungen, funktioniert die Konsole für Benutzer, die eine Rolle mit dieser IAM-Richtlinie haben, nicht wie vorgesehen. Um sicherzustellen, dass diese Benutzer die CodeDeploy Konsole weiterhin verwenden können, fügen Sie die `AWSCodeDeployReadOnlyAccess` verwaltete Richtlinie außerdem der Rolle hinzu, die dem Benutzer zugewiesen wurde, wie unter beschrieben. AWS verwaltete (vordefinierte) Richtlinien für CodeDeploy

Sie müssen Benutzern, die nur die API AWS CLI oder die CodeDeploy API aufrufen, keine Mindestberechtigungen für die Konsole gewähren.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder 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": "*"
        }
    ]
}
```

Fehlerbehebung AWS CodeDeploy Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit CodeDeploy und IAM auftreten können.

Themen

- Ich bin nicht berechtigt, IAM durchzuführen: PassRole
- Ich möchte Personen außerhalb meines AWS Konto, um auf meine CodeDeploy Ressourcen zuzugreifen

Ich bin nicht berechtigt, IAM durchzuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an CodeDeploy übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in CodeDeploy auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS Konto, um auf meine CodeDeploy Ressourcen zuzugreifen

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen CodeDeploy unterstützt werden, finden Sie unterWie AWS CodeDeploy funktioniert mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

CodeDeploy Referenz zu Berechtigungen

Verwenden Sie die folgende Tabelle, wenn Sie Zugriffs- und Schreibberechtigungsrichtlinien einrichten, die Sie einer IAM-Identität zuordnen können (identitätsbasierte Richtlinien). In der Tabelle sind alle CodeDeploy API-Operationen, die Aktionen, für die Sie Berechtigungen zur Ausführung der Aktion erteilen können, und das Format des Ressourcen-ARN aufgeführt, der für die Erteilung von Berechtigungen verwendet werden soll. Sie geben die Aktionen im Feld `Action` der Richtlinie an. Sie geben einen ARN mit oder ohne Platzhalterzeichen (*) als Ressourcenwert im Feld `Resource` der Richtlinie an.

Sie können in Ihren CodeDeploy Richtlinien AWS Bedingungsschlüssel für alle Bereiche verwenden, um Bedingungen auszudrücken. Eine vollständige Liste der AWS-weiten Schlüssel finden Sie unter Verfügbare Schlüssel im IAM-Benutzerhandbuch.

Um eine Aktion anzugeben, verwenden Sie das Präfix `codedeploy:` gefolgt vom Namen der API-Operation (z. B. `codedeploy:GetApplication` und `codedeploy:CreateApplication`). Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Komma (z. B. `"Action": ["codedeploy:action1", "codedeploy:action2"]`).

Verwenden von Platzhalterzeichen

Sie können ein Platzhalterzeichen (*) in dem ARN verwenden, um mehrere Aktionen oder Ressourcen anzugeben. `codedeploy:*`Gibt beispielsweise alle Aktionen an und gibt alle CodeDeploy Aktionen `codedeploy:Get*` an, die mit dem Wort beginnen. CodeDeploy `Get` Im folgenden Beispiel wird Zugriff auf alle Bereitstellungsgruppen mit Namen gewährt, die mit `West` beginnen und im Zusammenhang mit Anwendungen stehen, deren Namen mit `Test` beginnen.

```
arn:aws:codedeploy:us-west-2:444455556666:deploymentgroup:Test*/West*
```

Sie können Platzhalterzeichen für die in der folgenden Tabelle aufgeführten Ressourcen verwenden:

- *application-name*
- *deployment-group-name*
- *deployment-configuration-name*
- *instance-ID*

Platzhalter können nicht zusammen mit *region* oder *account-id* verwendet werden. Weitere Informationen zu Platzhaltern finden Sie unter IAM Identifiers im Benutzerhandbuch von IAM.

> **Note**
>
> Im ARN folgt bei jeder Aktion nach der Ressource ein Doppelpunkt (:). Sie können die Ressource auch mit einem Schrägstrich (/) verfolgen. Weitere Informationen finden Sie in den CodeDeployBeispiel-ARNs.

CodeDeploy API-Operationen und erforderliche Berechtigungen für Aktionen

AddTagsToOnPremisesInstances

Aktion: `codedeploy:AddTagsToOnPremisesInstances`

Tags müssen zu einer oder mehreren lokalen Instances hinzugefügt werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:instance/`*`instance-ID`*

BatchGetApplicationRevisions

Aktion: `codedeploy:BatchGetApplicationRevisions`

Informationen zu mehreren Anwendungsrevisionen im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

BatchGetApplications

Aktion: `codedeploy:BatchGetApplications`

Informationen zu mehreren Anwendungen im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:*`

BatchGetDeploymentGroups

Aktion: `codedeploy:BatchGetDeploymentGroups`

Informationen über mehrere Bereitstellungsgruppen im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

BatchGetDeploymentInstances

Aktion: `codedeploy:BatchGetDeploymentInstances`

Informationen über eine oder mehrere Instances, die Teil einer Bereitstellungsgruppe sind, müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

BatchGetDeployments

Aktion: `codedeploy:BatchGetDeployments`

Informationen über mehrere Bereitstellungen im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

BatchGetOnPremisesInstances

Aktion: `codedeploy:BatchGetOnPremisesInstances`

Informationen über eine oder mehrere lokale Instances müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:region:account-id:*`

ContinueDeployment

Aktion: `codedeploy:ContinueDeployment`

Während einer blue/green Bereitstellung erforderlich, um mit der Registrierung von Instances in einer Ersatzumgebung mit einem Elastic Load Balancing Load Balancer zu beginnen.

Ressource: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

CreateApplication

Aktion: `codedeploy:CreateApplication`

Es muss eine Anwendung im Zusammenhang mit dem -Benutzer erstellt werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

CreateDeployment[1]

Aktion: `codedeploy:CreateDeployment`

Es muss eine Bereitstellung für eine Anwendung im Zusammenhang mit dem -Benutzer erstellt werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

CreateDeploymentConfig

Aktion: `codedeploy:CreateDeploymentConfig`

Es muss eine benutzerdefinierte Bereitstellungskonfiguration im Zusammenhang mit dem -Benutzer erstellt werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentconfig/`*`deployment-configuration-name`*

CreateDeploymentGroup

Aktion: `codedeploy:CreateDeploymentGroup`

Es muss eine Bereitstellungsgruppe für eine Anwendung im Zusammenhang mit dem -Benutzer erstellt werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

DeleteApplication

Aktion: `codedeploy:DeleteApplication`

Es muss eine Anwendung im Zusammenhang mit dem -Benutzer gelöscht werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

DeleteDeploymentConfig

Aktion: `codedeploy:DeleteDeploymentConfig`

Es muss eine benutzerdefinierte Bereitstellungskonfiguration im Zusammenhang mit dem -Benutzer gelöscht werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentconfig/`*`deployment-configuration-name`*

DeleteDeploymentGroup

Aktion: `codedeploy:DeleteDeploymentGroup`

Es muss eine Bereitstellungsgruppe für eine Anwendung im Zusammenhang mit dem -Benutzer gelöscht werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

DeregisterOnPremisesInstance

Aktion: `codedeploy:DeregisterOnPremisesInstance`

Eine lokale Instance muss abgemeldet werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:instance/`*`instance-ID`*

GetApplication

Aktion: `codedeploy:GetApplication`

Informationen zu einer einzelnen Anwendung im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

GetApplicationRevision

Aktion: `codedeploy:GetApplicationRevision`

Informationen über eine einzelne Anwendungsrevision für eine Anwendung im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

GetDeployment

Aktion: `codedeploy:GetDeployment`

Informationen über eine einzelne Bereitstellung in einer Bereitstellungsgruppe für eine Anwendung im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

GetDeploymentConfig

Aktion: `codedeploy:GetDeploymentConfig`

Informationen zu einer einzelnen Bereitstellungskonfiguration im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentconfig/`*`deployment-configuration-name`*

GetDeploymentGroup

Aktion: `codedeploy:GetDeploymentGroup`

Informationen über eine einzelne Bereitstellungsgruppe für eine Anwendung im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

GetDeploymentInstance

Aktion: `codedeploy:GetDeploymentInstance`

Informationen zu einer einzelnen Instance in einer Bereitstellung im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

GetOnPremisesInstance

Aktion: `codedeploy:GetOnPremisesInstance`

Informationen über eine einzelne lokale Instance müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

ListApplicationRevisions

Aktion: `codedeploy:ListApplicationRevisions`

Informationen über alle Anwendungsrevisionen für eine Anwendung im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:region:account-id:application:*`

ListApplications

Aktion: `codedeploy:ListApplications`

Informationen zu allen Anwendungen im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:region:account-id:application:*`

ListDeploymentConfigs

Aktion: `codedeploy:ListDeploymentConfigs`

Informationen zu allen Bereitstellungskonfiguration im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:region:account-id:deploymentconfig/*`

ListDeploymentGroups

Aktion: `codedeploy:ListDeploymentGroups`

Informationen über alle Bereitstellungsgruppen für eine Anwendung im Zusammenhang mit dem -Benutzer müssen abgerufen werden.

Ressource: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/*`

ListDeploymentInstances

Aktion: `codedeploy:ListDeploymentInstances`

Erforderlich, um Informationen über alle Instanzen in einer Bereitstellung abzurufen, die dem Benutzer zugeordnet sind.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

ListDeployments

Aktion: `codedeploy:ListDeployments`

Erforderlich, um Informationen über alle Bereitstellungen in einer Bereitstellungsgruppe abzurufen, die dem Benutzer zugeordnet ist, oder um alle Bereitstellungen abzurufen, die dem Benutzer zugeordnet sind.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

ListGitHubAccountTokenNames

Aktion: `codedeploy:ListGitHubAccountTokenNames`

Erforderlich, um eine Liste der Namen der gespeicherten Verbindungen zu GitHub Konten abzurufen.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:*`

ListOnPremisesInstances

Aktion: `codedeploy:ListOnPremisesInstances`

Es muss eine Liste einer oder mehrerer Instance-Namen abgerufen werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:*`

RegisterApplicationRevision

Aktion: `codedeploy:RegisterApplicationRevision`

Informationen über eine Anwendungsrevision für eine Anwendung im Zusammenhang mit dem - Benutzer müssen registriert werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

RegisterOnPremisesInstance

Aktion: `codedeploy:RegisterOnPremisesInstance`

Erforderlich, um eine lokale Instanz bei CodeDeploy zu registrieren.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:instance/`*`instance-ID`*

RemoveTagsFromOnPremisesInstances

Aktion: `codedeploy:RemoveTagsFromOnPremisesInstances`

Tags müssen aus einer oder mehreren lokalen Instances entfernt werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:instance/`*`instance-ID`*

SkipWaitTimeForInstanceTermination

Aktion: `codedeploy:SkipWaitTimeForInstanceTermination`

Erforderlich, um eine angegebene Wartezeit zu überschreiben und sofort nach erfolgreicher Weiterleitung des Datenverkehrs in einer Bereitstellung mit dem Beenden von Instanzen in der ursprünglichen Umgebung zu beginnen. blue/green

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:instance/`*`instance-ID`*

StopDeployment

Aktion: `codedeploy:StopDeployment`

Eine laufende Bereitstellung in einer Bereitstellungsgruppe für eine Anwendung im Zusammenhang mit dem -Benutzer muss gestoppt werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

UpdateApplication[3]

Aktion: `codedeploy:UpdateApplication`

Informationen zu einer Anwendung im Zusammenhang mit dem -Benutzer müssen geändert werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

UpdateDeploymentGroup[3]

Aktion: `codedeploy:UpdateDeploymentGroup`

Informationen über eine einzelne Bereitstellungsgruppe für eine Anwendung im Zusammenhang mit dem -Benutzer müssen geändert werden.

Ressource: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

[1] Wenn Sie `CreateDeployment` Berechtigungen angeben, müssen Sie auch `GetDeploymentConfig` Berechtigungen für die Bereitstellungskonfiguration `GetApplicationRevision` und/oder `RegisterApplicationRevision` Berechtigungen für die Anwendungsversion angeben.

[2] Gültig, `ListDeployments` wenn Sie eine Bereitstellungsgruppe angeben, aber nicht, wenn Sie alle Bereitstellungen auflisten, die dem Benutzer zugeordnet sind.

[3] Für`UpdateApplication`: Sie müssen über `UpdateApplication` Berechtigungen sowohl für den alten als auch für den neuen Anwendungsnamen verfügen. Für `UpdateDeploymentGroup`-Aktionen, bei denen der Name einer Bereitstellungsgruppe geändert werden muss, müssen Sie über `UpdateDeploymentGroup`-Berechtigungen für den alten und den neuen Namen der Bereitstellungsgruppe verfügen.

Cross-service verwirrter Abgeordneter, Prävention

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. Im AWS Fall eines dienststellenübergreifenden Identitätswechsels kann es zu einem Problem mit verwirrten Stellvertretern kommen. Cross-service Ein Identitätswechsel kann auftreten, wenn ein Dienst (der anrufende Dienst) einen anderen Dienst (den angerufenen Dienst) aufruft. Der Anruf-Dienst kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, werden Tools AWS bereitgestellt, mit denen Sie Ihre Daten für alle Dienste mit Dienstprinzipalen schützen können, denen Zugriff auf Ressourcen in Ihrem Konto gewährt wurde.

Wir empfehlen die Verwendung der SourceAccount globalen Bedingungsschlüssel aws: SourceArn und aws: in Ressourcenrichtlinien, um die Berechtigungen einzuschränken, die CodeDeploy der Ressource einen anderen Dienst gewähren. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie diese

Option, `aws:SourceAccount` wenn Sie möchten, dass Ressourcen in diesem Konto mit der dienstübergreifenden Nutzung verknüpft werden.

Bei EC2/On-Premises AWS Lambda- und regulären Amazon ECS-Bereitstellungen `aws:SourceArn` sollte der Wert von den ARN der CodeDeploy Bereitstellungsgruppe enthalten, mit der die CodeDeploy IAM-Rolle übernommen werden darf.

Bei Amazon blue/green ECS-Bereitstellungen, die durch erstellt wurden CloudFormation, `aws:SourceArn` sollte der Wert von den CloudFormation Stack-ARN enthalten, mit dem die CodeDeploy IAM-Rolle übernommen werden darf.

Der effektivste Weg, sich vor dem Problem des verwirrten Stellvertreters zu schützen, besteht darin, den `aws:SourceArn` Schlüssel mit dem vollständigen ARN der Ressource zu verwenden. Wenn Sie den vollständigen ARN nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie Platzhalterzeichen (*) für die unbekannten Teile.

Sie könnten beispielsweise die folgende Vertrauensrichtlinie bei einer EC2/On-Premises AWS Lambda- oder regulären Amazon ECS-Bereitstellung verwenden:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "codedeploy.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:codedeploy:us-
east-1:111122223333:deploymentgroup:myApplication/*"
                }
            }
        }
    ]
```

```
}
```

Für eine Amazon blue/green ECS-Bereitstellung CloudFormation, die über erstellt wurde, können Sie Folgendes verwenden:

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/*"
                }
            }
        }
    ]
}
```

Anmeldung und Überwachung CodeDeploy

Dieser Abschnitt bietet eine Übersicht über die Überwachung, Protokollierung und Reaktion auf Vorfälle in CodeDeploy.

Prüfung aller Interaktionen mit CodeDeploy

CodeDeploy ist CodeDeploy in einen Dienst integriert AWS CloudTrail, der API-Aufrufe erfasst, die von oder im Namen Ihres AWS Kontos getätigt wurden, und die Protokolldateien an einen von

Ihnen angegebenen S3-Bucket übermittelt. CloudTrail erfasst API-Aufrufe von der CodeDeploy Konsole, von CodeDeploy Befehlen über die AWS CLI oder direkt von den CodeDeploy APIs aus. Anhand der von gesammelten Informationen können Sie feststellen CloudTrail, an welche Adresse die Anfrage gestellt wurde CodeDeploy, von welcher Quell-IP-Adresse die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde usw. Weitere Informationen CloudTrail dazu finden Sie unter Arbeiten mit CloudTrail Protokolldateien im AWS CloudTrail Benutzerhandbuch.

Sie können die durch eine CodeDeploy Bereitstellung erstellten Protokolldaten anzeigen, indem Sie den CloudWatch Amazon-Agenten so einrichten, dass er aggregierte Daten in der CloudWatch Konsole anzeigt, oder indem Sie sich bei einer Instance anmelden, um die Protokolldatei zu überprüfen. Weitere Informationen finden Sie unter CodeDeploy Agentenprotokolle senden an CloudWatch.

Alarmierungs- und Vorfallmanagement

Sie können Amazon CloudWatch Events verwenden, um Änderungen im Status einer Instance oder einer Bereitstellung (eines Ereignisses) in Ihrem CodeDeploy Betrieb zu erkennen und darauf zu reagieren. Auf der Grundlage der von Ihnen erstellten Regeln ruft CloudWatch Events dann eine oder mehrere Zielaktionen auf, wenn eine Bereitstellung oder Instance in den Zustand übergeht, den Sie in einer Regel angeben. Abhängig von der Statusänderung können Sie Benachrichtigungen versenden, Statusinformationen erfassen, Korrekturmaßnahmen ausführen, Ereignisse auslösen oder andere Aktionen ausführen. Sie können die folgenden Zieltypen auswählen, wenn Sie CloudWatch Ereignisse als Teil Ihrer CodeDeploy Operationen verwenden:

- AWS Lambda Funktionen
- Kinesis-Streams
- Amazon SQS SQS-Warteschlangen
- Built-in Ziele (Alarmaktionen) CloudWatch
- Amazon-SNS-Themen

Nachfolgend finden Sie einige Anwendungsfälle:

- Verwenden Sie eine Lambda-Funktion, um eine Benachrichtigung an einen Slack-Kanal zu senden, wenn eine Bereitstellung fehlschlägt.
- Push-Daten über Bereitstellungen oder Instances in einem Kinesis Stream unterstützen die umfassende Echtzeit-Statusüberwachung.

- Verwenden Sie CloudWatch Alarmaktionen, um EC2-Instances automatisch zu stoppen, zu beenden, neu zu starten oder wiederherzustellen, wenn ein von Ihnen festgelegtes Bereitstellungs- oder Instance-Ereignis eintritt.

Weitere Informationen finden Sie unter Was ist Amazon CloudWatch Events im CloudWatch Amazon-Benutzerhandbuch.

Konformitätsüberprüfung für AWS CodeDeploy

Unser neues AWS Anmeldeerlebnis ist nicht für regulierte Workloads konzipiert. Wenn Sie unser neues AWS Anmeldeformular nutzen, es aber AWS für regulierte Workloads nutzen möchten, können Sie sich für AWS (erweiterte) Funktionen registrieren oder erweiterte Funktionen für Ihre Umgebung aktivieren. AWS

Informationen darüber, ob AWS-Service ein in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter AWS-Services Umfang nach Compliance-Programm unter Umfang nach Compliance-Programm AWS-Services das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der AWS Sicherheitsdokumentation.

Resilienz in AWS CodeDeploy

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Sicherheit der Infrastruktur in AWS CodeDeploy

Als verwalteter Service AWS CodeDeploy ist er durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper Amazon Web Services: Sicherheitsprozesse im Überblick beschrieben sind.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff CodeDeploy über das Netzwerk. Clients müssen Transport Layer Security (TLS) 1.2 oder höher unterstützen. Wir empfehlen TLS 1.3 oder höher. Clients müssen außerdem Cipher Suites mit Perfect Forward Secrecy (PFS) wie Ephemeral (DHE) oder Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) unterstützen. Diffie-Hellman Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Anforderungen müssen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der mit einem IAM-Prinzipal verknüpft ist. Alternativ können Sie mit AWS -Security-Token-Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Referenz

Referenz.

Themen

- CodeDeploy AppSpec Dateiverweis
- CodeDeploy Referenz zur Agentenkonfiguration
- CloudFormation Vorlagen als CodeDeploy Referenz
- Verwendung CodeDeploy mit Amazon Virtual Private Cloud
- CodeDeploy Referenz zum Resource Kit
- CodeDeploy Kontingente

CodeDeploy AppSpec Dateiverweis

Dieser Abschnitt dient nur als Referenz. Einen konzeptionellen Überblick über die AppSpec Datei finden Sie unterApplication Specification Files.

Die Anwendungsspezifikationsdatei (AppSpec Datei) ist eine YAML-formatierte JSON-formatted Datei oder Datei, die CodeDeploy zur Verwaltung einer Bereitstellung verwendet wird.

> **Note**
>
> Die AppSpec Datei für eine EC2/On-Premises Bereitstellung muss benannt werden`appspec.yml`, es sei denn, Sie führen eine lokale Bereitstellung durch. Weitere Informationen finden Sie unter Erstellen Sie eine lokale Bereitstellung.

Themen

- AppSpec Dateien auf einer Amazon ECS-Rechenplattform
- AppSpec Dateien auf einem AWS Lambda Plattform für die Datenverarbeitung
- AppSpec Dateien auf einer EC2/on-premises Rechenplattform
- AppSpec Struktur der Datei
- AppSpec Beispiel für eine Datei
- AppSpec Abstand zwischen den Dateien

- Bestätigen Sie Ihre AppSpec Datei und den Speicherort

AppSpec Dateien auf einer Amazon ECS-Rechenplattform

Für Amazon ECS-Rechenplattformanwendungen wird die AppSpec Datei verwendet, um CodeDeploy Folgendes zu ermitteln:

- Ihre Amazon ECS-Aufgabendefinitionsdatei. Dies wird mit seinem ARN in der `TaskDefinition` Anweisung in der AppSpec Datei angegeben.
- Der Container und der Port in Ihrem Ersatzaufgabensatz, an den Ihr Application Load Balancer oder Network Load Balancer den Datenverkehr während einer Bereitstellung umleitet. Dies wird mit der `LoadBalancerInfo` Anweisung in der Datei angegeben. AppSpec
- Optionale Informationen über Ihren Amazon ECS-Service, z. B. die Plattformversion, auf der er ausgeführt wird, seine Subnetze und seine Sicherheitsgruppen.
- Optionale Lambda-Funktionen, die während Hooks ausgeführt werden, die Lebenszyklusereignissen während einer Amazon ECS-Bereitstellung entsprechen. Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“ für eine Amazon ECS-Bereitstellung.

AppSpec Dateien auf einem AWS Lambda Plattform für die Datenverarbeitung

Für AWS Lambda-Compute-Plattformanwendungen wird die AppSpec Datei verwendet, CodeDeploy um Folgendes zu ermitteln:

- Welche Lambda-Funktionsversion soll bereitgestellt werden.
- Welche Lambda-Funktionen sollen als Validierungstests verwendet werden?

Eine AppSpec Datei kann YAML-formatted oder JSON-formatted sein. Sie können den Inhalt einer AppSpec Datei auch direkt in die CodeDeploy Konsole eingeben, wenn Sie eine Einrichtung erstellen.

AppSpec Dateien auf einer EC2/on-premises Rechenplattform

Wenn Ihre Anwendung die EC2/On-Premises Rechenplattform verwendet, muss es sich bei der AppSpec Datei um eine YAML-formatted Datei mit einem Namen handeln `appspec.yml` und sie muss sich im Stammverzeichnis der Verzeichnisstruktur des Quellcodes einer Anwendung befinden.

Andernfalls schlagen Bereitstellungen fehl. Sie wird verwendet, um CodeDeploy Folgendes zu ermitteln:

- Was es auf Ihren Instances aus Ihrer Anwendungsversion in Amazon S3 oder installieren sollte GitHub.
- Welche Lebenszyklusereignis-Hooks als Reaktion auf Bereitstellungslebenszyklusereignisse ausgeführt werden sollen.

Nachdem Sie eine vollständige AppSpec Datei erstellt haben, bündeln Sie sie zusammen mit dem Inhalt, der bereitgestellt werden soll, in einer Archivdatei (zip, tar oder komprimiertes Tar). Weitere Informationen finden Sie unter Arbeiten mit Anwendungsrevisionen für CodeDeploy.

> **Note**
>
> Die Archivdateiformate tar und komprimiertes TAR-Archiv (.tar und .tar.gz) werden für Windows Server-Instanzen nicht unterstützt.

Nachdem Sie eine gebündelte Archivdatei (bekannt CodeDeploy als Revision) haben, laden Sie sie in einen Amazon S3 S3-Bucket oder ein Git-Repository hoch. Anschließend verwenden Sie CodeDeploy , um die Revision bereitzustellen. Detaillierte Anweisungen finden Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy.

Die Datei appspec.yml für eine EC2/On-Premises Compute-Plattform-Bereitstellung wird im Stammverzeichnis Ihrer Revision gespeichert. Weitere Informationen erhalten Sie unter Fügen Sie eine AppSpec Datei für eine EC2/On-Premises Bereitstellung hinzu und Planen Sie eine Überarbeitung für CodeDeploy.

AppSpec Struktur der Datei

Im Folgenden finden Sie die allgemeine Struktur für eine AppSpec Datei, die für Bereitstellungen auf AWS Lambda- und EC2/On-Premises Computerplattformen verwendet wird.

Ein Wert in einer YAML-formatted AppSpec Datei, bei dem es sich um eine Zeichenfolge handelt, darf nicht in Anführungszeichen („“) eingeschlossen werden, sofern nicht anders angegeben.

AppSpec Dateistruktur für Amazon ECS-Bereitstellungen

> **Note**
>
> Diese AppSpec Datei ist in YAML geschrieben, aber Sie können dieselbe Struktur verwenden, um eine Datei in JSON zu schreiben. Eine Zeichenfolge in einer JSON-formatted AppSpec Datei ist immer in Anführungszeichen („“) eingeschlossen.

```
version: 0.0
resources:
  ecs-service-specifications
hooks:
  deployment-lifecycle-event-mappings
```

In dieser Struktur:

Version

In diesem Abschnitt wird die Version der AppSpec Datei angegeben. Ändern Sie diesen Wert nicht. Sie ist erforderlich. Der einzige zulässige Wert ist derzeit **`0.0`**. Es ist CodeDeploy für die future Verwendung reserviert.

Geben Sie version mit einer Zeichenfolge an.

Ressourcen

Dieser Abschnitt enthält Informationen über die bereitzustellende Amazon ECS-Anwendung.

Weitere Informationen finden Sie unter AppSpec Abschnitt „Ressourcen“ für Amazon ECS-Bereitstellungen.

hooks

In diesem Abschnitt werden Lambda-Funktionen beschrieben, die bei bestimmten Event-Hooks für den Bereitstellungslebenszyklus ausgeführt werden, um die Bereitstellung zu validieren.

Weitere Informationen finden Sie unter Liste der Lifecycle-Event-Hooks für eine Amazon ECS-Bereitstellung.

AppSpec Dateistruktur für AWS Lambda-Bereitstellungen

> **Note**
>
> Diese AppSpec Datei ist in YAML geschrieben, aber Sie können dieselbe Struktur verwenden, um eine AppSpec Datei für eine Lambda-Bereitstellung in JSON zu schreiben. Eine Zeichenfolge in einer JSON-formatted AppSpec Datei ist immer in Anführungszeichen („“) eingeschlossen.

```
version: 0.0
resources:
  lambda-function-specifications
hooks:
  deployment-lifecycle-event-mappings
```

In dieser Struktur:

Version

In diesem Abschnitt wird die Version der AppSpec Datei angegeben. Ändern Sie diesen Wert nicht. Sie ist erforderlich. Der einzige zulässige Wert ist derzeit **`0.0`**. Es ist CodeDeploy für die future Verwendung reserviert.

Geben Sie version mit einer Zeichenfolge an.

Ressourcen

Dieser Abschnitt enthält Informationen über die bereitzustellende Lambda-Funktion.

Weitere Informationen finden Sie unter AppSpec Abschnitt „Ressourcen“ (Amazon ECS und AWS Lambda (nur Bereitstellungen).

hooks

In diesem Abschnitt werden Lambda-Funktionen beschrieben, die bei bestimmten Ereignissen im Bereitstellungslebenszyklus ausgeführt werden sollen, um die Bereitstellung zu validieren.

Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“.

AppSpec Dateistruktur für Bereitstellungen EC2/On-Premises

```
version: 0.0
os: operating-system-name
files:
  source-destination-files-mappings
permissions:
  permissions-specifications
hooks:
  deployment-lifecycle-event-mappings
```

In dieser Struktur:

Version

In diesem Abschnitt wird die Version der AppSpec Datei angegeben. Ändern Sie diesen Wert nicht. Sie ist erforderlich. Der einzige zulässige Wert ist derzeit **`0.0`**. Es ist CodeDeploy für die future Verwendung reserviert.

Geben Sie version mit einer Zeichenfolge an.

os

In diesem Abschnitt wird der Betriebssystemwert der Instance angegeben, auf der Sie bereitstellen. Es ist erforderlich. Die folgenden Werte können angegeben werden:

- linux — Die Instance ist eine Amazon Linux-, Ubuntu Server-, RHEL- oder macOS-Instance.
- windows — Die Instance ist eine Windows Server-Instance.

Geben Sie os mit einer Zeichenfolge an.

files

In diesem Abschnitt werden die Namen der Dateien angegeben, die während dem Install-Ereignis der Bereitstellung auf die Instance kopiert werden sollen.

Weitere Informationen finden Sie unter AppSpec Abschnitt „Dateien“ (nur EC2/On-Premises Bereitstellungen).

permissions

In diesem Abschnitt wird angegeben, wie spezielle Berechtigungen, sofern vorhanden, auf die Dateien im Abschnitt `files` angewendet werden sollen, wenn diese auf die Instance kopiert

werden. Dieser Abschnitt gilt nur für Amazon Linux-, Ubuntu Server-, Red Hat Enterprise Linux (RHEL) und macOS-Instances.

Weitere Informationen finden Sie unter AppSpec Abschnitt „Berechtigungen“ (nur Bereitstellungen) EC2/On-Premises.

hooks

In diesem Abschnitt werden Skripts angegeben, die bei bestimmten Bereitstellungslebenszyklusereignissen während der Bereitstellung ausgeführt werden sollen.

Weitere Informationen finden Sie unter AppSpec Abschnitt „Hooks“.

Themen

- AppSpec Abschnitt „Dateien“ (nur EC2/On-Premises Bereitstellungen)
- AppSpec Abschnitt „Ressourcen“ (Amazon ECS und AWS Lambda (nur Bereitstellungen)
- AppSpec Abschnitt „Berechtigungen“ (nur Bereitstellungen) EC2/On-Premises
- AppSpec Abschnitt „Hooks“

AppSpec Abschnitt „Dateien“ (nur EC2/On-Premises Bereitstellungen)

Enthält Informationen CodeDeploy darüber, welche Dateien aus Ihrer Anwendungsversion während des Installationsereignisses auf der Instanz installiert werden sollten. Dieser Abschnitt ist nur erforderlich, wenn Sie während der Bereitstellung Dateien aus Ihrer Revision an Standorte auf der Instance kopieren.

Dieser Abschnitt hat die folgende Struktur:

```
files:
  - source: source-file-location-1
    destination: destination-file-location-1
file_exists_behavior: DISALLOW|OVERWRITE|RETAIN
```

Mehrere `source`- und `destination`-Paare können festgelegt werden.

Die `source`-Anweisung identifiziert eine Datei oder ein Verzeichnis aus Ihrer Revision, die bzw. das auf die Instance kopiert werden soll:

- Wenn sich `source` auf eine Datei bezieht, werden nur die angegebenen Dateien auf die Instance kopiert.
- Wenn sich `source` auf ein Verzeichnis bezieht, werden alle Dateien im Verzeichnis auf die Instance kopiert.
- Wenn `source` es sich um einen einzelnen Schrägstrich handelt („/“ für Amazon Linux-, RHEL- und Ubuntu-Server-Instances oder „\“ für Windows Server-Instances), dann werden alle Dateien aus Ihrer Revision in die Instance kopiert.

Die in verwendeten Pfade `source` beziehen sich auf die `appspec.yml` Datei, die sich im Stammverzeichnis Ihrer Revision befinden sollte. Einzelheiten zur Dateistruktur einer Revision finden Sie unterPlanen Sie eine Überarbeitung für CodeDeploy.

Die `destination`-Anweisung identifiziert den Standort auf der Instance, an den die Dateien kopiert werden sollen. Dies muss ein vollständig qualifizierter Pfad sein, z. B. `/root/destination/directory` (unter Linux, RHEL und Ubuntu) oder `c:\destination\folder` (unter Windows).

`source` und `destination` werden jeweils mit einer Zeichenfolge angegeben.

Die `file_exists_behavior` Anweisung ist optional und gibt an, wie CodeDeploy mit Dateien umgegangen wird, die bereits an einem Bereitstellungszielort vorhanden sind, aber nicht Teil der vorherigen erfolgreichen Bereitstellung waren. Diese Einstellung kann einen der folgenden Werte annehmen:

- DISALLOW: Die Bereitstellung schlägt fehl. Dies ist auch das Standardverhalten, wenn keine Option angegeben ist.
- ÜBERSCHREIBEN: Die Version der Datei aus der Anwendungsrevision, die gerade bereitgestellt wird, ersetzt die Version, die sich bereits auf der Instanz befindet.
- RETAIN: Die Version der Datei, die sich bereits auf der Instanz befindet, wird beibehalten und als Teil der neuen Bereitstellung verwendet.

Wenn Sie die `file_exists_behavior` Einstellung verwenden, sollten Sie sich darüber im Klaren sein, dass diese Einstellung:

- kann nur einmal angegeben werden und gilt für alle Dateien und Verzeichnisse, die unter aufgeführt sind`files:`.
- hat Vorrang vor der `--file-exists-behavior` AWS CLI Option und der `fileExistsBehavior` API-Option (beide sind ebenfalls optional).

Hier ist ein `files` Beispielabschnitt für eine Amazon Linux-, Ubuntu Server- oder RHEL-Instance.

```
files:
  - source: Config/config.txt
    destination: /webapps/Config
  - source: source
    destination: /webapps/myApp
```

In diesem Beispiel werden die folgenden beiden Operationen während des Installations-Ereignisses ausgeführt:

1. Kopieren Sie die Datei `Config/config.txt` in Ihrer Revision zum `/webapps/Config/config.txt`-Pfad auf der Instance.
2. Kopieren Sie alle Dateien im `source`-Verzeichnis Ihrer Revision rekursiv in das Verzeichnis `/webapps/myApp` auf der Instance.

Beispiele für den Abschnitt „Dateien“

Die folgenden Beispiele zeigen, wie Sie den Abschnitt `files` angeben. Diese Beispiele beschreiben zwar die Datei- und Verzeichnisstrukturen (Ordner) von Windows Server, können aber problemlos für Amazon Linux-, Ubuntu Server- und RHEL-Instances angepasst werden.

> **Note**
>
> Nur EC2/On-Premises Bereitstellungen verwenden den `files` Abschnitt. Sie gilt nicht für AWS Lambda-Bereitstellungen.

Für die folgenden Beispiele nehmen wir an, dass diese Dateien im Bündel im Stammverzeichnis von `source` angezeigt werden:

- `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
```

Für die folgenden Beispielen nehmen wir an, dass `appspec.yml` im Bündel im Stammverzeichnis von `source` zusammen mit einem Ordner mit dem Namen `my-folder`, der drei Dateien enthält, angezeigt wird:

- `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 Abschnitt „Ressourcen“ (Amazon ECS und AWS Lambda (nur Bereitstellungen)

Der Inhalt des `'resources'` Abschnitts der AppSpec Datei hängt von der Rechenplattform Ihrer Bereitstellung ab. Der `'resources'` Abschnitt für eine Amazon ECS-Bereitstellung enthält Ihre Amazon ECS-Aufgabendefinition, den Container und den Port für die Weiterleitung des Datenverkehrs an Ihren aktualisierten Amazon ECS-Aufgabensatz sowie weitere optionale Informationen. Der `'resources'` Abschnitt für eine AWS Lambda Bereitstellung enthält den Namen, den Alias, die aktuelle Version und die Zielversion einer Lambda-Funktion.

Themen

- AppSpec Abschnitt „Ressourcen“ für AWS Lambda-Bereitstellungen
- AppSpec Abschnitt „Ressourcen“ für Amazon ECS-Bereitstellungen

AppSpec Abschnitt „Ressourcen“ für AWS Lambda-Bereitstellungen

Der `'resources'` Abschnitt spezifiziert die bereitzustellende Lambda-Funktion und hat die folgende Struktur:

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"
            }
        }
    }
]
```

Jede Eigenschaft wird mit einer Zeichenfolge angegeben.

- `name` – Erforderlich. Dies ist der Name der Lambda-Funktion, die bereitgestellt werden soll.
- `alias` – Erforderlich. Dies ist der Name des Alias für die Lambda-Funktion.
- `currentversion` – Erforderlich. Dies ist die Version der Lambda-Funktion, auf die der Traffic derzeit verweist. Dieser Wert muss eine gültige positive Ganzzahl sein.
- `targetversion` – Erforderlich. Dies ist die Version der Lambda-Funktion, auf die der Datenverkehr umgestellt wird. Dieser Wert muss eine gültige positive Ganzzahl sein.

AppSpec Abschnitt „Ressourcen“ für Amazon ECS-Bereitstellungen

Der `'resources'` Abschnitt spezifiziert den bereitzustellenden Amazon ECS-Service und hat die folgende Struktur:

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
                }
            ]
        }
    }
}
]
```

Jede Eigenschaft wird mit einer Zeichenfolge angegeben, mit Ausnahme von`ContainerPort`, bei der es sich um eine Zahl handelt.

- `TaskDefinition` – Erforderlich. Dies ist die Aufgabendefinition für den bereitzustellenden Amazon ECS-Service. Sie wird mit dem ARN der Aufgabendefinition angegeben. Das ARN-Format ist `arn:aws:ecs:`*`aws-region`*`:`*`account-id`*`:task-definition/`*`task-definition-family`*`:`*`task-definition-revision`*. Weitere Informationen finden Sie unter Amazon Resource Names (ARNs) und AWS Service Namespaces.

  > **Note**
  >
  > Der `:`*`task-definition-revision`* Teil des ARN ist optional. Wenn es weggelassen wird, verwendet Amazon ECS die neueste ACTIVE-Version der Aufgabendefinition.

- `ContainerName` – Erforderlich. Dies ist der Name des Amazon ECS-Containers, der Ihre Amazon ECS-Anwendung enthält. Es muss sich um einen Container handeln, der in Ihrer Amazon ECS-Aufgabendefinition angegeben ist.
- `ContainerPort` – Erforderlich. Dies ist der Port auf dem Container, zu dem der Verkehr weitergeleitet wird.
- `PlatformVersion`: Optional. Die Plattformversion der Fargate-Aufgaben im bereitgestellten Amazon ECS-Service. Weitere Informationen finden Sie unter AWS Fargate -Plattformversionen. Wenn nicht angegeben, `LATEST` wird sie standardmäßig verwendet.
- `NetworkConfiguration`: Optional. Unter `AwsvpcConfiguration` können Sie Folgendes angeben. Weitere Informationen finden Sie AwsVpcConfigurationin der Amazon ECS Container Service API-Referenz.
  - `Subnets`: Optional. Eine durch Kommas getrennte Liste von einem oder mehreren Subnetzen in Ihrem Amazon ECS-Service.
  - `SecurityGroups`: Optional. Eine durch Kommas getrennte Liste mit einer oder mehreren Sicherheitsgruppen in Ihrem Amazon Elastic Container Service.

- `AssignPublicIp`: Optional. Eine Zeichenfolge, die angibt, ob die elastic network interface Ihres Amazon ECS-Service eine öffentliche IP-Adresse erhält. Die gültigen Werte sind `ENABLED` und `DISABLED`.

> **Note**
>
> Alle oder keine der Einstellungen unter `NetworkConfiguration` müssen angegeben werden. Wenn Sie beispielsweise `Subnets` angeben möchten, müssen Sie auch `SecurityGroups` und `AssignPublicIp` angeben. Wenn keine angegeben ist, werden die aktuellen Netzwerkeinstellungen von Amazon ECS CodeDeploy verwendet.

- `CapacityProviderStrategy`: Optional. Eine Liste der Amazon ECS-Kapazitätsanbieter, die Sie für Ihre Bereitstellung verwenden möchten. Weitere Informationen finden Sie unter Amazon ECS Capacity Providers im Amazon Elastic Container Service Developer Guide. Für jeden Kapazitätsanbieter können Sie die folgenden Einstellungen angeben. Einzelheiten zu diesen Einstellungen finden Sie unter AWS: :ECS:: ServiceCapacityProviderStrategyItem im AWS CloudFormation Benutzerhandbuch
  - `Base`: Optional. Der Basiswert gibt an, wie viele Aufgaben mindestens mit dem angegebenen Kapazitätsanbieter ausgeführt werden sollen. In einer Kapazitätsanbieter-Strategie kann nur für einen Kapazitätsanbieter ein Basiswert festgelegt werden. Wenn kein Wert angegeben wird, wird der Standardwert 0 verwendet.
  - `CapacityProvider`: Optional. Der Kurzname des Kapazitätsanbieters. Beispiel: capacityProviderA
  - `Weight`: Optional.

    Der Gewichtungswert gibt den relativen Prozentsatz der Gesamtzahl der gestarteten Aufgaben an, die den angegebenen Kapazitätsanbieter verwenden sollten. Der `weight`-Wert wird berücksichtigt, nachdem der `base`-Wert, falls definiert, erfüllt ist.

    Wenn kein `weight`-Wert angegeben wird, wird der Standardwert `0` verwendet. Wenn mehrere Capacity Provider innerhalb einer Capacity Provider-Strategie angegeben werden, muss mindestens einer der Capacity Provider einen Gewichtungswert von mehr als Null haben, und alle Capacity Provider mit einem Gewicht von `0` werden nicht für Aufgaben verwendet. Wenn Sie mehrere Capacity Provider in einer Strategie angeben, die alle ein Gewichtung von `0` haben, schlagen alle `RunTask`- oder `CreateService`-Aktionen, die die Capacity-Provider-Strategie verwenden, fehl.

Ein Beispielszenario für die Verwendung von Gewichtungen ist die Definition einer Strategie, die zwei Kapazitätsanbieter enthält, wobei beide eine Gewichtung von 1 haben. Wenn die `base` erfüllt ist, werden die Aufgaben gleichmäßig auf die beiden Kapazitätsanbieter aufgeteilt. Wenn Sie unter Anwendung derselben Logik eine Gewichtung von 1 für Kapazitätsanbieter A und eine Gewichtung von 4 für Kapazitätsanbieter B, angeben, würden für jede Aufgabe, die mit Kapazitätsanbieter A, ausgeführt wird, vier Aufgaben mit Kapazitätsanbieter B ausgeführt.

AppSpec Abschnitt „Berechtigungen“ (nur Bereitstellungen) EC2/On-Premises

`'permissions'` In diesem Abschnitt wird festgelegt, wie spezielle Berechtigungen, falls vorhanden, auf die Dateien und auf den `'files'` Abschnitt angewendet werden sollen, nachdem sie directories/folders in die Instanz kopiert wurden. Sie können mehrere `object`-Anweisungen angeben. Dieser Abschnitt ist optional. Sie gilt nur für Amazon Linux-, Ubuntu Server-, RHEL- und macOS-Instances.

> **Note**
>
> Auf macOS-Instanzen werden die `context` Anweisungen `acls` und nicht unterstützt. Eine Bereitstellung, deren AppSpec Datei eine von beiden auf macOS verwendet, schlägt fehl, bevor Dateien installiert wurden. Die anderen Anweisungen funktionieren wie in diesem Abschnitt beschrieben.

> **Note**
>
> Der `'permissions'`-Abschnitt wird nur für EC2/On-Premises-Bereitstellungen verwendet. Es wird nicht für AWS Lambda- oder Amazon ECS-Bereitstellungen verwendet.

Dieser Abschnitt hat die folgende Struktur:

```
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
```

Folgende Anleitungen sind zu beachten:

- `object` – Erforderlich. Dabei handelt es sich um eine Gruppe von Dateisystemobjekten (Dateien oder directories/folders), auf die die angegebenen Berechtigungen angewendet werden, nachdem die Dateisystemobjekte in die Instance kopiert wurden.

  Geben Sie `object` mit einer Zeichenfolge an.

- `pattern` – Optional. Gibt ein Muster für das Anwenden von Berechtigungen an. Wenn nichts angegeben ist oder wenn mit den Sonderzeichen **"**"** angegeben, werden die Berechtigungen abhängig von `type` auf alle übereinstimmenden Dateien oder Verzeichnisse angewendet.

  Geben Sie `pattern` mit einer Zeichenfolge in Anführungszeichen ("") an.

- `except` – Optional. Gibt alle Dateien oder Verzeichnisse an, die Ausnahmen für `pattern` sind.

  Geben Sie `except` mit einer durch Komma getrennten Liste von Zeichenfolgen in eckigen Klammern an.

- `owner` – Optional. Der Name des Eigentümers von `object`. Wenn nicht angegeben, bleiben alle vorhandenen Besitzer, die auf die ursprüngliche Datei oder directory/folder Struktur angewendet wurden, nach dem Kopiervorgang unverändert.

  Geben Sie `owner` mit einer Zeichenfolge an.

- `group` – Optional. Der Name der Gruppe für `object`. Wenn nicht angegeben, bleiben alle vorhandenen Gruppen, die auf die Originaldatei oder directory/folder Struktur angewendet wurden, nach dem Kopiervorgang unverändert.

  Geben Sie `group` mit einer Zeichenfolge an.

- `mode` – Optional. Ein numerischer Wert, der die Berechtigungen angibt, auf die angewendet werden sollobject. Die Moduseinstellung folgt der Linux-Befehlssyntax chmod.

> **Important**
>
> Wenn der Wert eine führende Null enthält, müssen Sie ihn mit doppelten Anführungszeichen umgeben oder die führende Null entfernen, sodass nur noch drei Ziffern übrig bleiben.

> **Note**
>
> Symbolische Schreibweise wie **`u+x`** wird für die `mode` Einstellung nicht unterstützt.

Beispiele:

- `mode: "0644"`erteilt dem Besitzer des Objekts Lese- und Schreibberechtigungen (6), Schreibberechtigungen für die Gruppe (4) und Schreibberechtigungen für alle anderen Benutzer (4).
- `mode: 644`gewährt dieselben Berechtigungen wie. `mode: "0644"`
- `mode: 4755`setzt das setuid-Attribut (4), gewährt dem Besitzer Vollzugriff (7), erteilt der Gruppe Lese- und Ausführungsrechte (5) und erteilt allen anderen Benutzern Lese- und Ausführungsberechtigungen (5).

  Weitere Beispiele finden Sie in der Dokumentation zum Linux-Befehl chmod.

  Wenn der Modus nicht angegeben ist, bleiben alle vorhandenen Modi, die auf die ursprüngliche Datei- oder Ordnerstruktur angewendet wurden, nach dem Kopiervorgang unverändert.

- `acls` – Optional. Eine Liste von Zeichenfolgen, die für einen oder mehrere Zugriffskontrolllisten (Access Control List, ACL)-Einträge stehen, die auf `object` angewendet werden. Beispielsweise steht **`u:bob:rw`** für Lese- und Schreibberechtigungen für den Benutzer **`bob`**. (Weitere Beispiele finden Sie unter den Beispielen für ACL-Eintragsformate in der Dokumentation zum Linux-Befehl `setfacl`.) Sie können mehrere ACL-Einträge angeben. Wenn nicht `acls` angegeben, bleiben alle vorhandenen ACLs, die auf die ursprüngliche Datei oder directory/folder Struktur angewendet wurden, nach dem Kopiervorgang unverändert. Diese ersetzen alle vorhandenen ACLs.

  Geben Sie einen `acls` mit einem Bindestrich (-), gefolgt von einem Leerzeichen, und dann einer Zeichenfolge (z. B. `- u:jane:rw`) an. Wenn Sie über mehr als eine ACL verfügen, wird jede jeweils in einer separaten Zeile angegeben.

> **Note**
>
> Wenn Sie unbenannte Benutzer, unbenannte Gruppen oder andere ähnliche ACL-Einträge festlegen, schlägt die AppSpec Datei fehl. Verwenden Sie stattdessen `mode`, um diese Art von Berechtigungen festzulegen.

- `context` – Optional. Bei Security-Enhanced Linux (SELinux) -fähigen Instanzen eine Liste sicherheitsrelevanter Kontext-Labels, die auf die kopierten Objekte angewendet werden sollen. Bezeichnungen werden als Schlüssel angegeben, die `user`, `type` und `range` enthalten. (Weitere Informationen finden Sie in der SELinux-Dokumentation.) Jeder Schlüssel wird mit einer Zeichenfolge eingegeben. Wenn nicht angegeben, bleiben alle vorhandenen Labels, die auf die Originaldatei oder directory/folder Struktur angewendet wurden, nach dem Kopiervorgang unverändert.
  - `user` – Optional. Der SELinux-Benutzer.
  - `type` – Optional. Der Name des SELinux-Typs.
  - `range` – Optional. Der SELinux-Bereichsspezifizierer. Dies hat keine Auswirkung, es sei denn, Multi-Level Sicherheit (MLS) und Multi-Category Sicherheit (MCS) sind auf dem Computer aktiviert. Wenn diese Option nicht aktiviert ist, ist der `range`-Standardwert **`s0`**.

  Geben Sie `context` mit einer Zeichenfolge an (z. B. `user: unconfined_u`). Jeder `context` wird in einer separaten Zeile angegeben.
- `type` – Optional. Die Arten von Objekten, auf die die angegebenen Berechtigungen angewendet werden sollen. `type` ist eine Zeichenfolge, die auf **`file`** oder **`directory`** eingestellt sein kann. Wenn **`file`** angegeben ist, werden die Berechtigungen nur auf Dateien angewendet, die sofort nach dem Kopiervorgang in `object` enthalten sind (und nicht auf `object` selbst). Wenn **`directory`** angegeben, werden die Berechtigungen rekursiv auf alle Rechte angewendet directories/folders , die sich `object` nach dem Kopiervorgang irgendwo befinden (aber nicht auf sich `object` selbst).

  Geben Sie `type` mit einem Bindestrich (-), gefolgt von einem Leerzeichen, und dann einer Zeichenfolge (z. B. `- file`) an.

Beispiel für den Abschnitt „Berechtigungen“

Das folgende Beispiel zeigt, wie Sie den `'permissions'`-Abschnitt mit den Anweisungen `object`, `pattern`, `except`, `owner`, `mode` und `type` angeben. Dieses Beispiel gilt nur für Amazon Linux-,

Ubuntu Server- und RHEL-Instances. In diesem Beispiel wird davon ausgegangen, dass die folgenden Dateien und Ordner auf die Instance in dieser Hierarchie kopiert werden:

```
/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
```

Die folgende AppSpec Datei zeigt, wie Sie Berechtigungen für diese Dateien und Ordner festlegen, nachdem sie kopiert wurden:

```
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
```

Es ergeben sich folgende Berechtigungen:

```
-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
```

Das folgende Beispiel zeigt, wie Sie den `'permissions'`-Abschnitt unter Hinzunahme der Anweisungen `acls` und `context` angeben. Dieses Beispiel gilt nur für Amazon Linux-, Ubuntu Server- und RHEL-Instances.

```
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 Abschnitt „Hooks“

Der Inhalt des `'hooks'` Abschnitts der AppSpec Datei variiert je nach Rechenplattform für Ihre Bereitstellung. Der Abschnitt `'hooks'` einer EC2/On-Premises -Bereitstellung enthält Zuweisungen, die Bereitstellungslebenszyklusereignis-Hooks mit einem oder mehreren Skripts verknüpfen. Der `'hooks'` Abschnitt für eine Lambda- oder Amazon ECS-Bereitstellung spezifiziert Lambda-Validierungsfunktionen, die während eines Bereitstellungslebenszyklus ausgeführt werden sollen. Wenn ein Ereignis-Hook nicht vorhanden ist, wird kein Vorgang für dieses Ereignis ausgeführt. Dieser Abschnitt ist nur erforderlich, wenn Sie im Rahmen der Bereitstellung Skripts oder Lambda-Validierungsfunktionen ausführen.

Themen

- AppSpec Abschnitt „Hooks“ für eine Amazon ECS-Bereitstellung
- AppSpec Abschnitt „Hooks“ für einen AWS Lambda-Bereitstellung
- AppSpec Abschnitt „Hooks“ für eine Bereitstellung EC2/On-Premises

AppSpec Abschnitt „Hooks“ für eine Amazon ECS-Bereitstellung

Themen

- Liste der Lifecycle-Event-Hooks für eine Amazon ECS-Bereitstellung
- Führen Sie die Reihenfolge der Hooks in einer Amazon ECS-Bereitstellung aus.
- Struktur des Abschnitts „Hooks“
- Beispiel für die Lambda-Funktion „Hooks“

Liste der Lifecycle-Event-Hooks für eine Amazon ECS-Bereitstellung

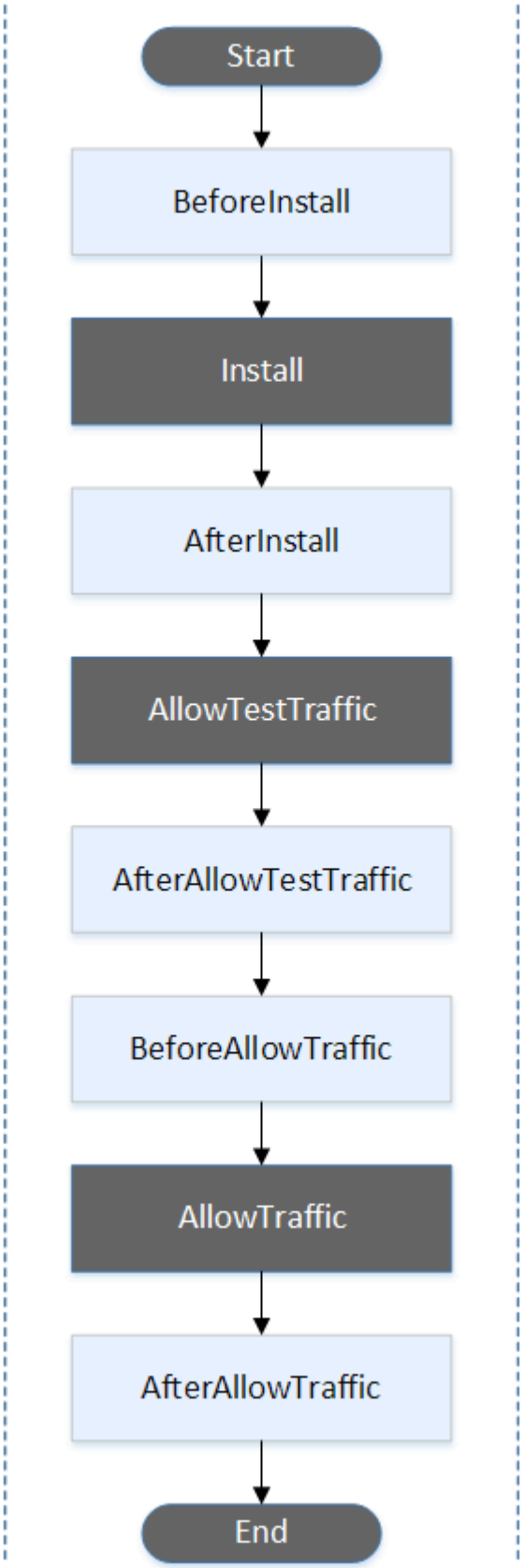
Ein AWS Lambda-Hook ist eine Lambda-Funktion, die mit einer Zeichenfolge in einer neuen Zeile nach dem Namen des Lebenszyklusereignisses angegeben wird. Jeder Hook wird einmal pro Bereitstellung ausgeführt. Im Folgenden finden Sie Beschreibungen der Lebenszyklusereignisse, bei denen Sie während einer Amazon ECS-Bereitstellung einen Hook ausführen können.

- `BeforeInstall`— Wird verwendet, um Aufgaben auszuführen, bevor der Ersatzaufgabensatz erstellt wird. Eine Zielgruppe wird dem ursprünglichen Aufgabesatz zugeordnet. Wenn ein optionaler Test-Listener angegeben wird, wird er dem ursprünglichen Aufgabensatz zugeordnet. Ein Rollback ist zu diesem Zeitpunkt nicht möglich.
- `AfterInstall`— Wird verwendet, um Aufgaben auszuführen, nachdem der Ersatz-Tasksatz erstellt wurde und ihm eine der Zielgruppen zugeordnet wurde. Wenn ein optionaler Test-Listener angegeben wird, wird er dem ursprünglichen Aufgabensatz zugeordnet. Die Ergebnisse einer hook-Funktion bei diesem Lebenszyklus-Ereignis können einen Rollback auslösen.
- `AfterAllowTestTraffic`— Wird verwendet, um Aufgaben auszuführen, nachdem der Test-Listener Traffic an den Ersatz-Tasksatz weitergeleitet hat. Die Ergebnisse einer hook-Funktion können zu diesem Zeitpunkt einen Rollback auslösen.
- `BeforeAllowTraffic`— Wird verwendet, um Aufgaben auszuführen, nachdem die zweite Zielgruppe dem Ersatz-Task-Set zugeordnet wurde, aber bevor der Verkehr auf den Ersatz-Tasksatz umgeleitet wird. Die Ergebnisse einer hook-Funktion bei diesem Lebenszyklus-Ereignis können einen Rollback auslösen.
- `AfterAllowTraffic`— Wird verwendet, um Aufgaben auszuführen, nachdem die zweite Zielgruppe Traffic an den Ersatz-Tasksatz weitergeleitet hat. Die Ergebnisse einer hook-Funktion bei diesem Lebenszyklus-Ereignis können einen Rollback auslösen.

Weitere Informationen erhalten Sie unter Was passiert während einer Amazon ECS-Bereitstellung und Tutorial: Bereitstellen eines Amazon ECS-Service mit einem Validierungstest.

Führen Sie die Reihenfolge der Hooks in einer Amazon ECS-Bereitstellung aus.

In einer Amazon ECS-Bereitstellung werden Event-Hooks in der folgenden Reihenfolge ausgeführt:


Start
BeforeInstall
Install
AfterInstall
AllowTestTraffic
AfterAllowTestTraffic
BeforeAllowTraffic
AllowTraffic
AfterAllowTraffic
End

> **Note**
>
> Für die Ereignisse Start TestTraffic, Install AllowTraffic, und End in der Bereitstellung können keine Skripts erstellt werden, weshalb sie in diesem Diagramm grau dargestellt werden.

Struktur des Abschnitts „Hooks“

Die folgenden Beispiele veranschaulichen die Struktur des Abschnitts `'hooks'`.

Mit YAML:

```
Hooks:
  - BeforeInstall: "BeforeInstallHookFunctionName"
  - AfterInstall: "AfterInstallHookFunctionName"
  - AfterAllowTestTraffic: "AfterAllowTestTrafficHookFunctionName"
  - BeforeAllowTraffic: "BeforeAllowTrafficHookFunctionName"
  - AfterAllowTraffic: "AfterAllowTrafficHookFunctionName"
```

Mit JSON:

```
"Hooks": [
  {
   "BeforeInstall": "BeforeInstallHookFunctionName"
  },
  {
   "AfterInstall": "AfterInstallHookFunctionName"
  },
  {
   "AfterAllowTestTraffic": "AfterAllowTestTrafficHookFunctionName"
  },
  {
   "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
  },
  {
   "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
  }
 ]
}
```

Beispiel für die Lambda-Funktion „Hooks“

Verwenden Sie den `'hooks'` Abschnitt, um eine Lambda-Funktion anzugeben, die aufgerufen CodeDeploy werden kann, um eine Amazon ECS-Bereitstellung zu validieren. Sie können dieselbe oder eine andere Funktion für die Ereignisse im Lebenszyklus der `AfterAllowTraffic` Bereitstellung `BeforeInstall` `AfterInstallAfterAllowTestTraffic`,`BeforeAllowTraffic`,, und verwenden. Nach Abschluss der Validierungstests ruft die `AfterAllowTraffic` Lambda-Funktion zurück CodeDeploy und liefert das Ergebnis `Succeeded` oder`Failed`.

> **Important**
>
> Die Bereitstellung gilt als fehlgeschlagen, wenn sie nicht innerhalb einer Stunde von der Lambda-Validierungsfunktion benachrichtigt CodeDeploy wird.

Vor dem Aufrufen einer Lambda-Hook-Funktion muss der Server mithilfe des Befehls über die Bereitstellungs-ID und die Ausführungs-ID des Lifecycle-Event-Hooks informiert werden. `putLifecycleEventHookExecutionStatus`

Im Folgenden finden Sie ein Beispiel für eine Lambda-Hook-Funktion, in Node.js die geschrieben wurde.

```
'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 Abschnitt „Hooks“ für einen AWS Lambda-Bereitstellung

Themen

- Liste der Lifecycle-Event-Hooks für einen AWS Lambda-Bereitstellung
- Reihenfolge der Hooks in einer Bereitstellung einer Lambda-Funktionsversion ausführen
- Struktur des Abschnitts „Hooks“
- Beispiel für die Lambda-Funktion „Hooks“

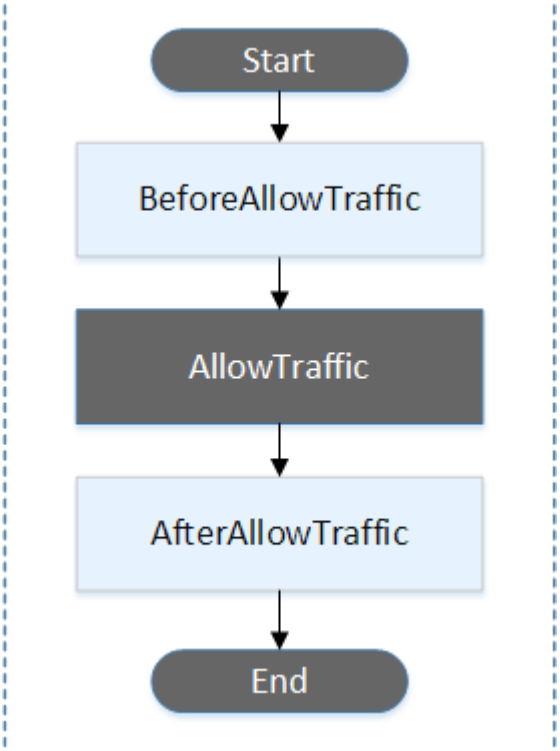
Liste der Lifecycle-Event-Hooks für einen AWS Lambda-Bereitstellung

Ein AWS Lambda-Hook ist eine Lambda-Funktion, die mit einer Zeichenfolge in einer neuen Zeile nach dem Namen des Lebenszyklusereignisses angegeben wird. Jeder Hook wird einmal pro Bereitstellung ausgeführt. Im Folgenden finden Sie Beschreibungen der Hooks, die in Ihrer AppSpec Datei verwendet werden können.

- BeforeAllowTraffic— Wird verwendet, um Aufgaben auszuführen, bevor der Datenverkehr auf die bereitgestellte Lambda-Funktionsversion umgestellt wird.
- AfterAllowTraffic— Wird verwendet, um Aufgaben auszuführen, nachdem der gesamte Datenverkehr auf die bereitgestellte Lambda-Funktionsversion umgestellt wurde.

Reihenfolge der Hooks in einer Bereitstellung einer Lambda-Funktionsversion ausführen

In einer serverlosen Bereitstellung einer Lambda-Funktionsversion werden Event-Hooks in der folgenden Reihenfolge ausgeführt:



> **Note**
>
> Für die Start AllowTraffic- und Endereignisse in der Bereitstellung kann kein Skript erstellt werden, weshalb sie in diesem Diagramm grau dargestellt werden.

Struktur des Abschnitts „Hooks“

Die folgenden Beispiele veranschaulichen die Struktur des 'hooks'-Abschnitts .

Mit YAML:

```
hooks:
   - BeforeAllowTraffic: BeforeAllowTrafficHookFunctionName
   - AfterAllowTraffic: AfterAllowTrafficHookFunctionName
```

Mit JSON:

```
"hooks": [{
    "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
    },
    {
    "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
}]
```

Beispiel für die Lambda-Funktion „Hooks“

Verwenden Sie den Abschnitt „Hooks“, um eine Lambda-Funktion anzugeben, die aufgerufen CodeDeploy werden kann, um eine Lambda-Bereitstellung zu validieren. Sie können dieselbe oder eine andere Funktion für die Ereignisse im Lebenszyklus der `BeforeAllowTraffic` `AfterAllowTraffic` Bereitstellung verwenden. Nach Abschluss der Validierungstests ruft die Lambda-Validierungsfunktion zurück CodeDeploy und liefert ein Ergebnis von `Succeeded` oder`Failed`.

> **Important**
>
> Die Bereitstellung gilt als fehlgeschlagen, wenn sie nicht innerhalb einer Stunde von der Lambda-Validierungsfunktion benachrichtigt CodeDeploy wird.

Vor dem Aufrufen einer Lambda-Hook-Funktion muss der Server mithilfe des Befehls über die Bereitstellungs-ID und die Ausführungs-ID des Lifecycle-Event-Hooks informiert werden. `putLifecycleEventHookExecutionStatus`

Im Folgenden finden Sie ein Beispiel für eine Lambda-Hook-Funktion, in Node.js die geschrieben wurde.

```
'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 Abschnitt „Hooks“ für eine Bereitstellung EC2/On-Premises

Themen

- Liste der Lifecycle-Event-Hooks
- Verfügbarkeit von Hooks für Lebenszyklus-Ereignisse
- Führt die Reihenfolge der Hooks in einem Deployment aus
- Struktur des Abschnitts „Hooks“
- Referenzieren von Dateien in Ihren Hook-Skripten
- Verfügbarkeit von Umgebungsvariablen für Hooks
- Beispiel für Hooks

Liste der Lifecycle-Event-Hooks

Ein EC2/On-Premises Deployment-Hook wird einmal pro Deployment für eine Instanz ausgeführt. Sie können ein oder mehrere Skripts zur Ausführung in einem Hook angeben. Jeder Hook für ein

Lebenszyklusereignis wird mit einer Zeichenfolge in einer separaten Zeile angegeben. Hier finden Sie Beschreibungen der Hooks, die für die Verwendung in Ihrer AppSpec Datei verfügbar sind.

Weitere Informationen darüber, welche Lebenszyklusereignis-Hooks für welche Bereitstellungs- und Rollback-Typen gültig sind, finden Sie unter Verfügbarkeit von Hooks für Lebenszyklus-Ereignisse.

- `ApplicationStop`— Dieses Ereignis im Bereitstellungslebenszyklus tritt auf, noch bevor die Anwendungsversion heruntergeladen wird. Sie können Skripts für dieses Ereignis angeben, um die Anwendung ordnungsgemäß anzuhalten, oder zur Vorbereitung einer Bereitstellung derzeit installierte Pakete entfernen. Die für dieses Ereignis im Bereitstellungslebenszyklus verwendeten AppSpec Dateien und Skripts stammen aus der vorherigen erfolgreich bereitgestellten Anwendungsversion.

  > **Note**
  >
  > Eine AppSpec Datei ist auf einer Instanz nicht vorhanden, bevor Sie sie auf ihr bereitstellen. Aus diesem Grund wird der `ApplicationStop`-Hook beim ersten Bereitstellen auf der Instance nicht ausgeführt. Sie können den `ApplicationStop`-Hook bei der zweiten Bereitstellung für eine Instance verwenden.

  Um den Speicherort der letzten erfolgreich bereitgestellten Anwendungsrevision zu ermitteln, sucht der CodeDeploy Agent nach dem in der *`deployment-group-id`*`_last_successful_install` Datei aufgeführten Speicherort. Diese Datei befindet sich unter:

  `/opt/codedeploy-agent/deployment-root/deployment-instructions`Ordner auf Amazon Linux-, Ubuntu Server- und RHEL Amazon EC2 EC2-Instances.

  `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions`Ordner auf Windows Server Amazon EC2 EC2-Instances.

  Informationen zur Problembehebung für eine Bereitstellung, die während des Bereitstellungslebenszyklusereignisses `ApplicationStop` fehlschlägt, finden Sie unter Behebung eines Fehlers oder eines ApplicationStop BeforeBlockTraffic Ereignisses im Lebenszyklus einer AfterBlockTraffic Bereitstellung.
- `DownloadBundle`— Während dieses Bereitstellungslebenszyklus kopiert der CodeDeploy Agent die Revisionsdateien der Anwendung an einen temporären Speicherort:

`/opt/codedeploy-agent/deployment-root/`*`deployment-group-id`*`/`*`deployment-id`*`/deployment-archive`Ordner auf Amazon Linux-, Ubuntu Server- und RHEL Amazon EC2 EC2-Instances.

`C:\ProgramData\Amazon\CodeDeploy\`*`deployment-group-id`*`\`*`deployment-id`*`\deployment-archive`Ordner auf Windows Server Amazon EC2 EC2-Instances.

Dieses Ereignis ist für den CodeDeploy Agenten reserviert und kann nicht zum Ausführen von Skripten verwendet werden.

Informationen zur Problembehebung für eine Bereitstellung, die während des Bereitstellungslebenszyklusereignisses `DownloadBundle` fehlschlägt, finden Sie unter Behebung eines fehlgeschlagenen DownloadBundle Deployment-Lifecycle-Ereignisses mit UnknownError: nicht zum Lesen geöffnet.

- `BeforeInstall`— Sie können dieses Ereignis im Bereitstellungslebenszyklus für Aufgaben vor der Installation verwenden, z. B. für das Entschlüsseln von Dateien und das Erstellen einer Sicherungskopie der aktuellen Version.
- `Install`— Während dieses Bereitstellungslebenszyklus kopiert der CodeDeploy Agent die Revisionsdateien vom temporären Speicherort in den endgültigen Zielordner. Dieses Ereignis ist für den CodeDeploy Agenten reserviert und kann nicht zum Ausführen von Skripten verwendet werden.
- `AfterInstall`— Sie können dieses Ereignis im Bereitstellungslebenszyklus für Aufgaben wie die Konfiguration Ihrer Anwendung oder das Ändern von Dateiberechtigungen verwenden.
- `ApplicationStart`— In der Regel verwenden Sie dieses Ereignis im Bereitstellungslebenszyklus, um Dienste neu zu starten, die währenddessen gestoppt wurden`ApplicationStop`.
- `ValidateService`— Dies ist das letzte Ereignis im Bereitstellungslebenszyklus. Es wird verwendet, um zu überprüfen, ob die Bereitstellung erfolgreich abgeschlossen wurde.
- `BeforeBlockTraffic`— Sie können dieses Ereignis im Bereitstellungslebenszyklus verwenden, um Aufgaben auf Instances auszuführen, bevor diese bei einem Load Balancer deregistriert werden.

  Informationen zur Problembehebung für eine Bereitstellung, die während des Bereitstellungslebenszyklusereignisses `BeforeBlockTraffic` fehlschlägt, finden Sie unter Behebung eines Fehlers oder eines ApplicationStop BeforeBlockTraffic Ereignisses im Lebenszyklus einer AfterBlockTraffic Bereitstellung.

- `BlockTraffic`— Während dieses Deployment-Lifecycle-Ereignisses wird der Internetverkehr daran gehindert, auf Instances zuzugreifen, die derzeit Traffic bereitstellen. Dieses Ereignis ist für den CodeDeploy Agenten reserviert und kann nicht zur Ausführung von Skripten verwendet werden.
- `AfterBlockTraffic`— Sie können dieses Ereignis im Bereitstellungslebenszyklus verwenden, um Aufgaben auf Instances auszuführen, nachdem diese bei ihrem jeweiligen Load Balancer abgemeldet wurden.

  Informationen zur Problembehebung für eine Bereitstellung, die während des Bereitstellungslebenszyklusereignisses `AfterBlockTraffic` fehlschlägt, finden Sie unter Behebung eines Fehlers oder eines ApplicationStop BeforeBlockTraffic Ereignisses im Lebenszyklus einer AfterBlockTraffic Bereitstellung.
- `BeforeAllowTraffic`— Sie können dieses Deployment-Lifecycle-Ereignis verwenden, um Aufgaben auf Instances auszuführen, bevor diese bei einem Load Balancer registriert wurden.
- `AllowTraffic`— Während dieses Bereitstellungslebenszyklus darf Internet-Traffic nach einer Bereitstellung auf Instances zugreifen. Dieses Ereignis ist für den CodeDeploy Agenten reserviert und kann nicht zur Ausführung von Skripten verwendet werden.
- `AfterAllowTraffic`— Sie können dieses Deployment-Lifecycle-Ereignis verwenden, um Aufgaben auf Instances auszuführen, nachdem diese bei einem Load Balancer registriert wurden.

Verfügbarkeit von Hooks für Lebenszyklus-Ereignisse

In der folgenden Tabelle finden Sie die Lebenszyklusereignis-Hooks für die einzelnen Bereitstellungs- und Rollback-Szenarien.

Name des Lebenszyklusereignis	Bereitstellung von Auto Scaling Scaling-Start[1]	Bereitstellung von Auto Scaling-Termination[1]	In-place Bereitstellung[2]	Blue/green Bereitstellung: Originalinstanzen	Blue/green Bereitstellung: Ersatzinstanzen	Blue/green Rollback der Bereitstellung: Originalinstanzen	Blue/green Rollback bei der Bereitstellung: Ersatzinstanzen
ApplicationStop	✓	✓	✓		✓		

Name des Lebenszyklusereignis	Bereitstellung von Auto Scaling Scaling-Start[1]	Bereitstellung von Auto Scaling-Termination[1]	In-place Bereitstellung[2]	Blue/green Bereitstellung: Originalinstanzen	Blue/green Bereitstellung: Ersatzinstanzen	Blue/green Rollback der Bereitstellung: Originalinstanzen	Blue/green Rollback bei der Bereitstellung: Ersatzinstanzen
DownloadBundle[3]	✓		✓		✓		
BeforeInstall	✓		✓		✓		
[3] installieren	✓		✓		✓		
AfterInstall	✓		✓		✓		
ApplicationStart	✓		✓		✓		
ValidateService	✓		✓		✓		
BeforeBlockTraffic		✓	✓	✓			✓
BlockTraffic[3]		✓	✓	✓			✓
AfterBlockTraffic		✓	✓	✓			✓
BeforeAllowTraffic	✓		✓		✓	✓	

Name des Lebenszyklusereignis	Bereitstellung von Auto Scaling Scaling-Start[1]	Bereitstellung von Auto Scaling-Termination[1]	In-place Bereitstellung[2]	Blue/green Bereitstellung: Originalinstanzen	Blue/green Bereitstellung: Ersatzinstanzen	Blue/green Rollback der Bereitstellung: Originalinstanzen	Blue/green Rollback bei der Bereitstellung: Ersatzinstanzen
AllowTraffic[3]	✓		✓		✓	✓	
AfterAllowTraffic	✓		✓		✓	✓	

[1] Informationen zu Amazon EC2 Auto Scaling Scaling-Bereitstellungen finden Sie unter. So funktioniert Amazon EC2 Auto Scaling mit CodeDeploy

[2] Gilt auch für das Rollback einer In-Place-Bereitstellung.

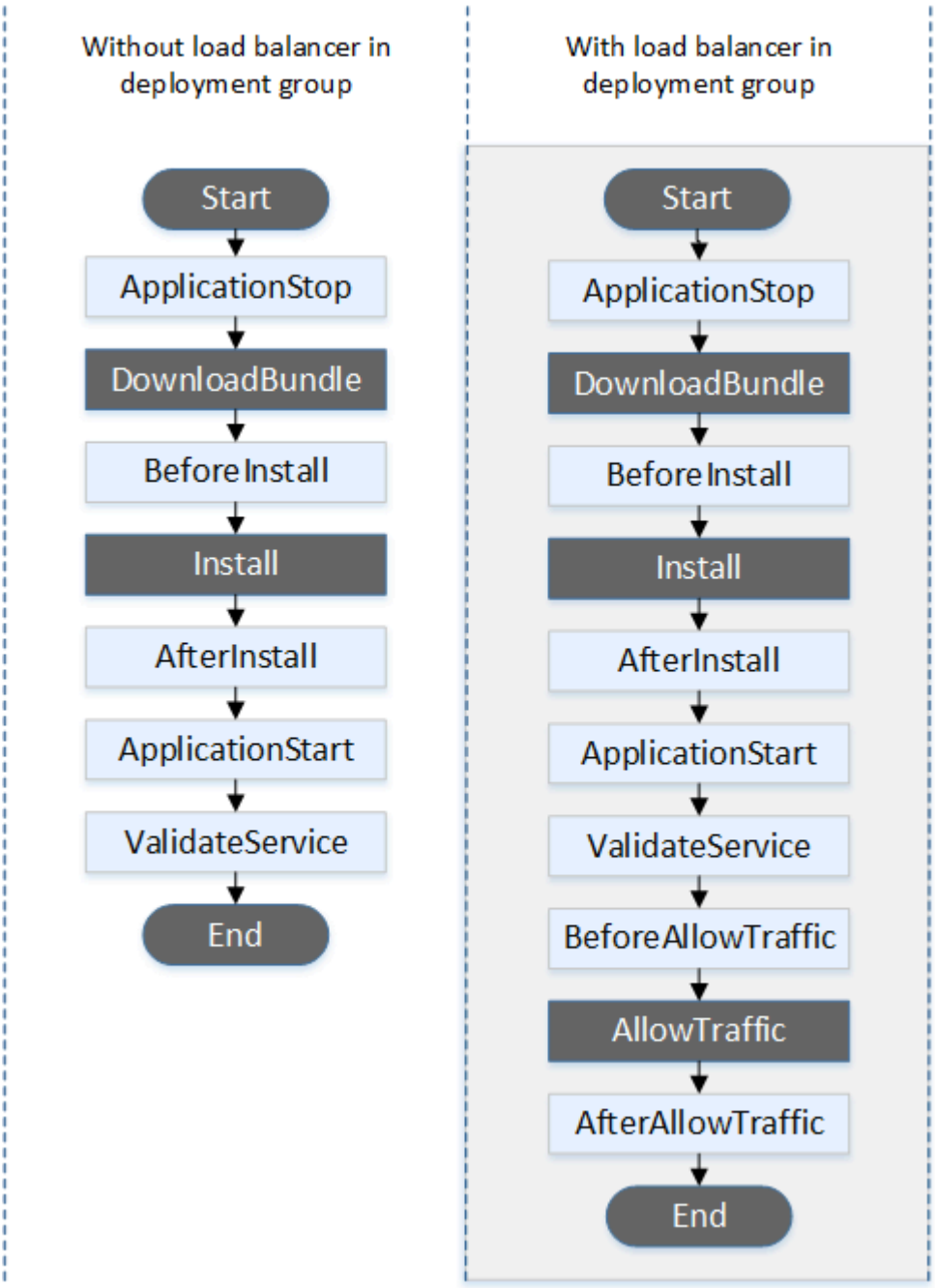
[3] Für den Betrieb reserviert CodeDeploy . Kann nicht für die Ausführung von Skripts verwendet werden.

Führt die Reihenfolge der Hooks in einem Deployment aus

Auto Scaling Scaling-Startbereitstellungen

CodeDeploy Führt während einer Auto Scaling Scaling-Startbereitstellung Event-Hooks in der folgenden Reihenfolge aus.

Weitere Informationen zu Auto Scaling Scaling-Startbereitstellungen finden Sie unterSo funktioniert Amazon EC2 Auto Scaling mit CodeDeploy.

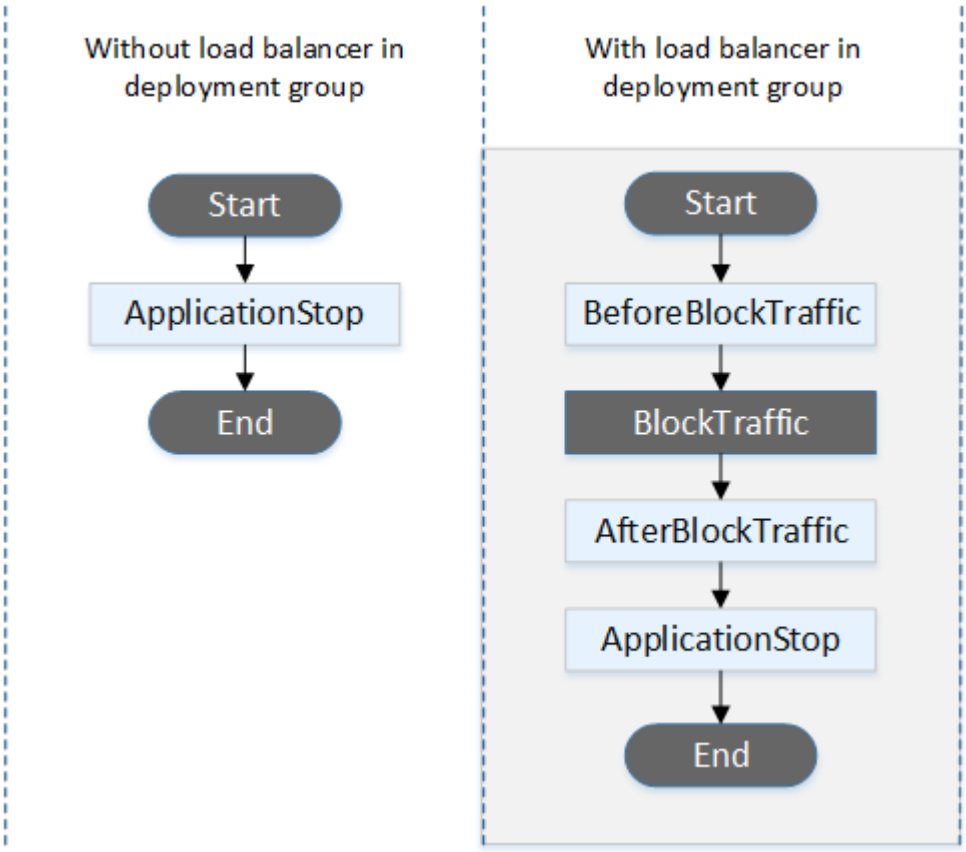


> **Note**
>
> Für die Start - DownloadBundleAllowTraffic, Installations - und Endereignisse in der Bereitstellung kann kein Skript erstellt werden, weshalb sie in diesem Diagramm grau dargestellt werden. Sie können den `'files'` Abschnitt der AppSpec Datei jedoch bearbeiten, um anzugeben, was während des Installationsereignisses installiert wird.

Bereitstellungen zur Kündigung von Auto Scaling

CodeDeploy Führt während einer Auto Scaling-Terminierungsbereitstellung Event-Hooks in der folgenden Reihenfolge aus.

Weitere Informationen zu Auto Scaling-Terminierungsbereitstellungen finden Sie unter[Aktivierung von Terminierungsbereitstellungen bei Auto Scaling-Scale-In-Ereignissen](#).

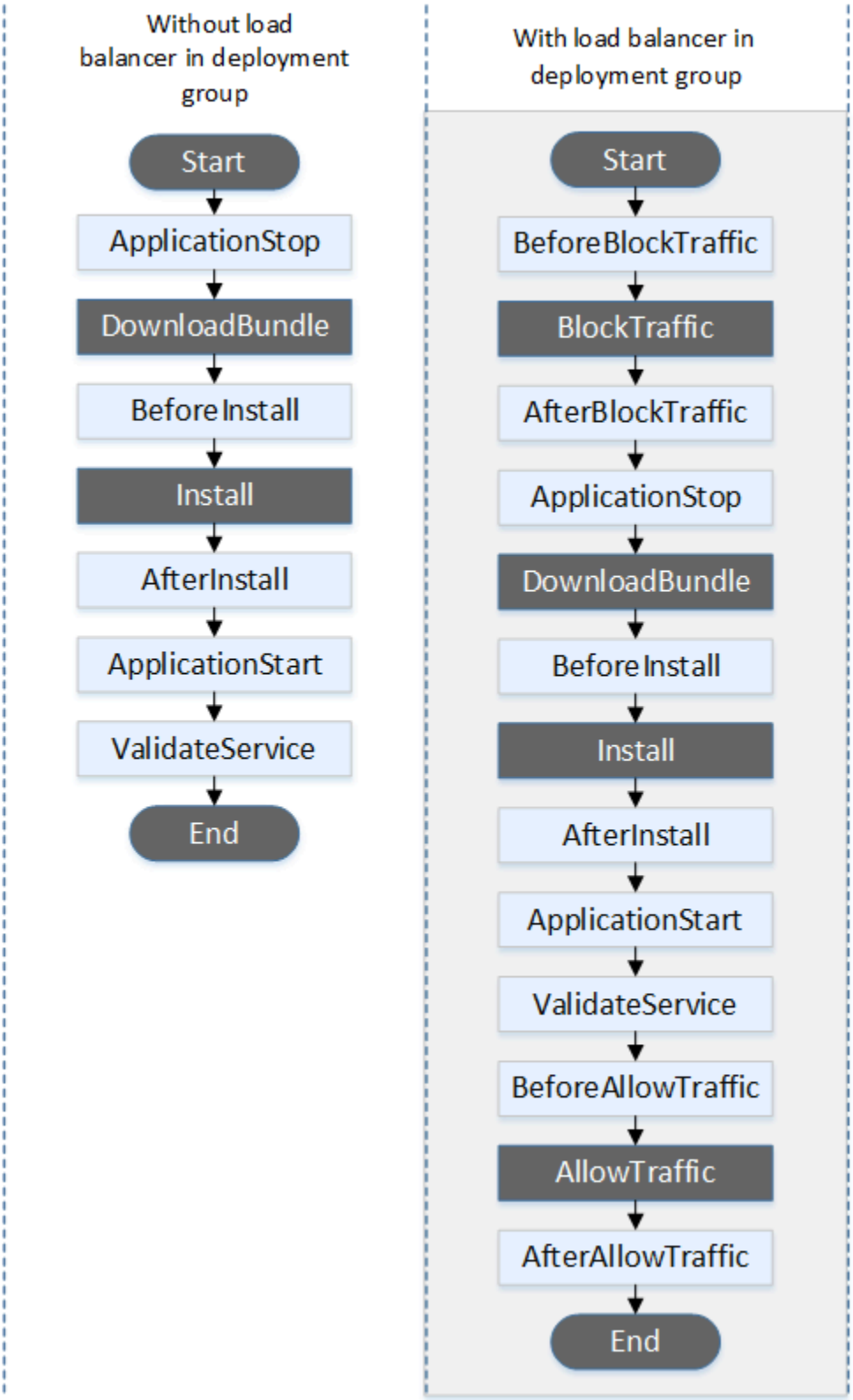


> **Note**
>
> Für die Start BlockTraffic- und Endereignisse in der Bereitstellung kann kein Skript erstellt werden, weshalb sie in diesem Diagramm grau dargestellt werden.

In-place Bereitstellungen

In einer In-Situ-Bereitstellung, einschließlich des Rollbacks einer In-Situ-Bereitstellung, werden Ereignis-Hooks in der folgenden Reihenfolge ausgeführt:

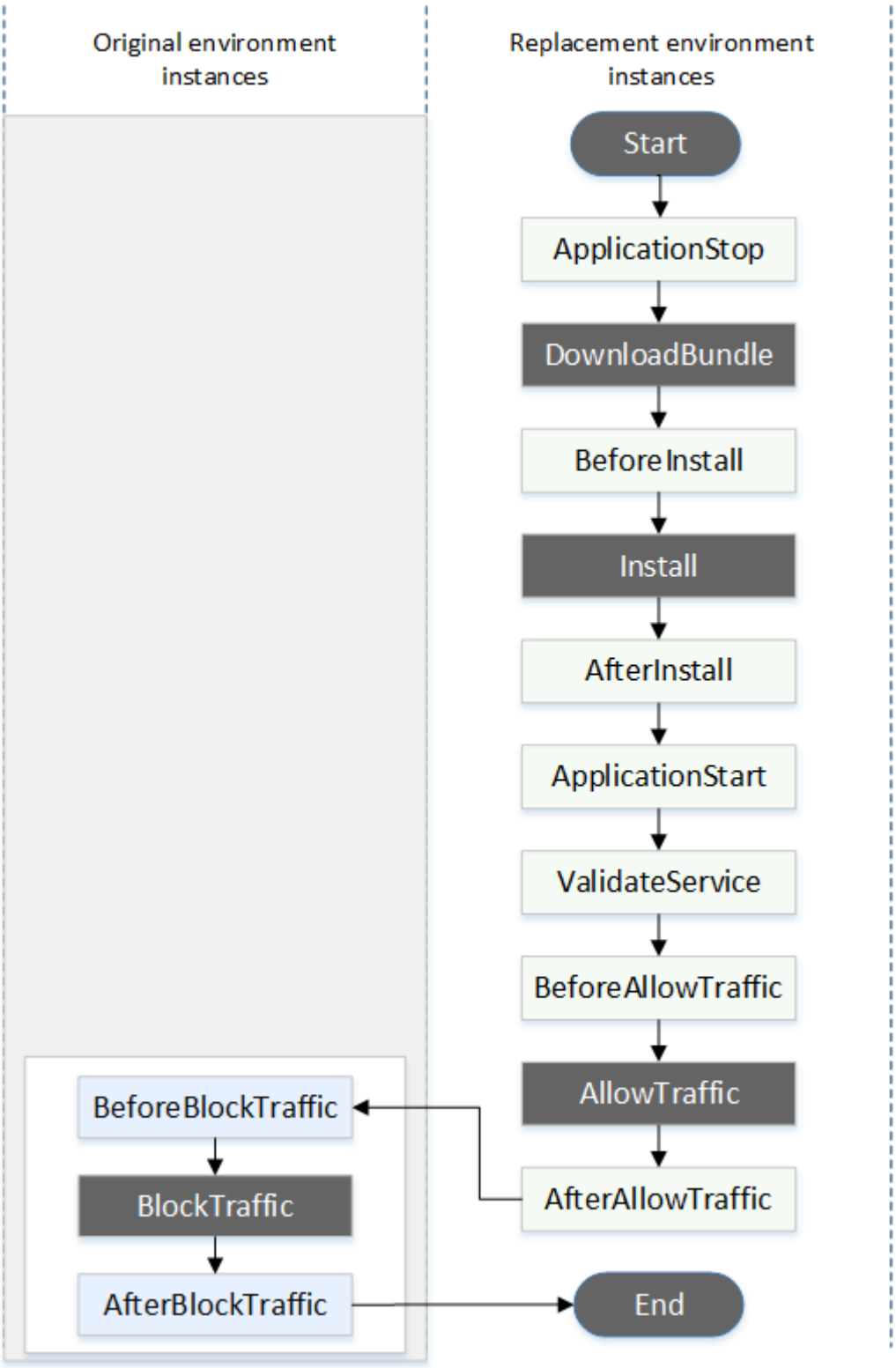
> **Note**
>
> Bei In-Place-Bereitstellungen gelten die sechs Hooks, die sich auf das Blockieren und Zulassen von Traffic beziehen, nur, wenn Sie in der Bereitstellungsgruppe einen Classic Load Balancer, Application Load Balancer oder Network Load Balancer von Elastic Load Balancing angeben.


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**
>
> Für die Start - DownloadBundle, Installations - und Endereignisse in der Bereitstellung kann kein Skript erstellt werden, weshalb sie in diesem Diagramm grau dargestellt werden. Sie können den `'files'` Abschnitt der AppSpec Datei jedoch bearbeiten, um anzugeben, was während des Installationsereignisses installiert wird.

Blue/green Bereitstellungen

In einer blue/green Bereitstellung werden Event-Hooks in der folgenden Reihenfolge ausgeführt:


Original environment instances
Replacement environment instances
Start
ApplicationStop
DownloadBundle
BeforeInstall
Install
AfterInstall
ApplicationStart
ValidateService
BeforeAllowTraffic
AllowTraffic
AfterAllowTraffic
BeforeBlockTraffic
BlockTraffic
AfterBlockTraffic
End

> **Note**
>
> Für die Start - DownloadBundle, Installations - BlockTrafficAllowTraffic, und Endereignisse in der Bereitstellung kann kein Skript erstellt werden, weshalb sie in diesem Diagramm grau dargestellt werden. Sie können jedoch den Abschnitt „Dateien“ der AppSpec Datei bearbeiten, um anzugeben, was während des Installationsereignisses installiert werden soll.

Struktur des Abschnitts „Hooks“

Der `'hooks'`-Abschnitt weist die folgende Struktur auf:

```
hooks:
   deployment-lifecycle-event-name:
     - location: script-location
       timeout: timeout-in-seconds
       runas: user-name
```

Sie können die folgenden Elemente nach dem Namen des Bereitstellungslebenszyklusereignisses in einen Hook-Eintrag einschließen:

location

Erforderlich Die Position im Paket der Skriptdatei für die Revision. Der Speicherort der Skripts, die Sie in `hooks` diesem Abschnitt angeben, bezieht sich auf das Stammverzeichnis des Revisionspakets der Anwendung. Weitere Informationen finden Sie unter Planen Sie eine Überarbeitung für CodeDeploy.

timeout

Optional. Die Anzahl der Sekunden, die das Skript ausgeführt wird, bevor es als fehlgeschlagen gilt. Der Standard ist 3 600 Sekunden (1 Stunde).

> **Note**
>
> Für jedes Bereitstellungslebenszyklusereignis sind maximal 3 600 Sekunden (1 Stunde) für die Ausführung von Skripts zulässig. Wenn Skripts dieses Limit überschreiten, wird die Bereitstellung angehalten und die Bereitstellung auf der Instance schlägt fehl. Stellen Sie sicher, dass die für timeout angegebene Gesamtanzahl der Sekunden für alle Skripts in jedem Bereitstellungslebenszyklusereignis dieses Limit nicht überschreitet.

runas

Optional. Der Benutzer, der vorgetäuscht werden soll, wenn das Skript ausgeführt wird. Standardmäßig ist dies der CodeDeploy Agent, der auf der Instanz ausgeführt wird. CodeDeploy speichert keine Passwörter, sodass der Benutzer nicht imitiert werden kann, wenn der Runas-Benutzer ein Passwort benötigt. Dieses Element gilt nur für Amazon Linux- und Ubuntu Server-Instances.

Referenzieren von Dateien in Ihren Hook-Skripten

Wenn Sie ein Skript mit einem CodeDeploy Lebenszyklusereignis verknüpfen, wie unter beschriebenAppSpec Abschnitt „Hooks“, und Sie in Ihrem Skript auf eine Datei verweisen möchten (z. B.`helper.sh`), müssen Sie Folgendes angeben`helper.sh`:

- (Empfohlen) Ein absoluter Pfad. Siehe Absolute Pfade verwenden.
- Ein relativer Pfad. Siehe Verwenden relativer Pfade.

Absolute Pfade verwenden

Um mit ihrem absoluten Pfad auf eine Datei zu verweisen, können Sie entweder:

- Geben Sie den absoluten Pfad im `files` Abschnitt der AppSpec Datei in der `destination` Eigenschaft an. Geben Sie dann denselben absoluten Pfad in Ihrem Hook-Skript an. Weitere Informationen finden Sie unter AppSpec Abschnitt „Dateien“ (nur EC2/On-Premises Bereitstellungen).
- Geben Sie einen dynamischen absoluten Pfad in Ihrem Hook-Skript an. Weitere Informationen finden Sie unter Speicherort des Deployment-Archivs.

Speicherort des Bereitstellungsarchivs

Während des DownloadBundleLebenszyklusereignisses extrahiert der CodeDeploy Agent die Version für die Bereitstellung in ein Verzeichnis, das das folgende Format hat:

`root-directory``/`*`deployment-group-id`*`/`*`deployment-id`*`/deployment-archive`

Der *`root-directory`* Teil des Pfads ist immer entweder auf den in der folgenden Tabelle angegebenen Standard festgelegt oder wird durch die `:root_dir` Konfigurationseinstellung

gesteuert. Weitere Informationen zu den Konfigurationseinstellungen finden Sie unterCodeDeploy Referenz zur Agentenkonfiguration.

Agentenplattform	Standard-Stammverzeichnis	
Linux — alle RPM-Distr ibutionen	`/opt/codedeploy-ag ent/deployment-root`	
Ubuntu Server — alle Deb-Distributionen	`/opt/codedeploy-ag ent/deployment-root`	
Windows Server	`%ProgramData%\Amazon \CodeDeploy`	

Von Ihren Hook-Skripten aus könnten Sie mithilfe des Stammverzeichnispfads und der `DEPLOYMENT_GROUP_ID` Umgebungsvariablen auf das `DEPLOYMENT_ID` aktuelle Deployment-Archiv zugreifen. Weitere Hinweise zu Variablen, die Sie verwenden können, finden Sie unterVerfügbarkeit von Umgebungsvariablen für Hooks.

So könnten Sie beispielsweise auf eine `data.json` Datei zugreifen, die sich im Stammverzeichnis Ihrer Revision unter Linux befindet:

```
#!/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)
```

Als weiteres Beispiel können Sie mit Powershell unter Windows auf eine `data.json` Datei zugreifen, die sich im Stammverzeichnis Ihrer Revision befindet:

```
$rootDirectory="$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)
```

Verwenden relativer Pfade

Um mit ihrem relativen Pfad auf eine Datei zu verweisen, müssen Sie das Arbeitsverzeichnis des CodeDeploy Agenten kennen. Dateipfade sind relativ zu diesem Verzeichnis.

Die folgende Tabelle zeigt das Arbeitsverzeichnis für jede unterstützte Plattform des CodeDeploy Agenten.

Agenten-Plattform	Methode zur Prozessverwaltung	Arbeitsverzeichnis für Lebenszyklusereignisskripte
Linux — alle RPM-Distributionen	systemd (Standard)	`/`
	init.d — Erfahre mehr	`/opt/codedeploy-agent`
Ubuntu Server — alle Debian-Distributionen	all	`/opt/codedeploy-agent`
Windows Server	Nicht zutreffend	`C:\Windows\System32`

Verfügbarkeit von Umgebungsvariablen für Hooks

Während jedes Bereitstellungslebenszyklusereignisses können Hook-Skripts auf die folgenden Umgebungsvariablen zugreifen:

APPLICATION_NAME

Der Name der Anwendung CodeDeploy , die Teil der aktuellen Bereitstellung ist (z. B.`WordPress_App`).

DEPLOYMENT_ID

Die ID, die der aktuellen Bereitstellung zugewiesen CodeDeploy wurde (z. B.`d-AB1CDEF23`).

DEPLOYMENT_GROUP_NAME

Der Name der Bereitstellungsgruppe CodeDeploy , die Teil der aktuellen Bereitstellung ist (z. B.`WordPress_DepGroup`).

DEPLOYMENT_GROUP_ID

Die ID der Bereitstellungsgruppe CodeDeploy , die Teil der aktuellen Bereitstellung ist (z. B.`b1a2189b-dd90-4ef5-8f40-4c1c5EXAMPLE`).

LIFECYCLE_EVENT

Der Name des aktuellen Bereitstellungslebenszyklusereignisses (z. B. `AfterInstall`).

Es handelt sich um lokale Umgebungsvariablen für die einzelnen Bereitstellungslebenszyklusereignisse.

Abhängig von der Quelle des Bereitstellungspakets stehen zusätzliche Umgebungsvariablen für Hook-Skripte zur Verfügung:

Paket von Amazon S3

- BUNDLE_BUCKET

  Der Name des Amazon S3 S3-Buckets, aus dem das Bereitstellungspaket heruntergeladen wurde (z. B.`my-s3-bucket`).
- BUNDLE_KEY

  Der Objektschlüssel für das heruntergeladene Paket innerhalb des Amazon S3 S3-Buckets (z. B.`WordPress_App.zip`).
- BUNDLE_VERSION

  Die Objektversion für das Bundle (zum Beispiel). `3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo` Diese Variable wird nur gesetzt, wenn für den Amazon S3 S3-Bucket die Objektversionierung aktiviert ist.
- BUNDLE_ETAG

  Das Objekt-Etag für das Bundle (zum Beispiel). `b10a8db164e0754105b7a99be72e3fe5-4`

Paket von GitHub

- BUNDLE_COMMIT

  Der SHA256-Commit-Hash des von Git generierten Bundles (zum Beispiel`d2a84f4b8b650937ec8f73cd8be2c74add5a911ba64df27458ed8229da804a26`).

Das folgende Skript ändert den Überwachungsport auf einem Apache HTTP-Server von 80 auf 9090, wenn der Wert von DEPLOYMENT_GROUP_NAME gleich `Staging` ist. Dieses Skript muss während des Bereitstellungslebenszyklusereignisses `BeforeInstall` aufgerufen werden:

```
if [ "$DEPLOYMENT_GROUP_NAME" == "Staging" ]
then
    sed -i -e 's/Listen 80/Listen 9090/g' /etc/httpd/conf/httpd.conf
fi
```

Das folgende Skriptbeispiel ändert die Ausführlichkeitsstufe der Nachrichten, die im Fehlerprotokoll aufgezeichnet werden, von Warnung zu Debug-Nachricht, wenn der Wert der Umgebungsvariablen DEPLOYMENT_GROUP_NAME gleich `Staging` ist. Dieses Skript muss während des Bereitstellungslebenszyklusereignisses `BeforeInstall` aufgerufen werden:

```
if [ "$DEPLOYMENT_GROUP_NAME" == "Staging" ]
then
    sed -i -e 's/LogLevel warn/LogLevel debug/g' /etc/httpd/conf/httpd.conf
fi
```

Das folgende Skriptbeispiel ersetzt den Text auf der angegebenen Webseite mit Text, der den Wert dieser Umgebungsvariablen anzeigt. Dieses Skript muss während des Bereitstellungslebenszyklusereignisses `AfterInstall` aufgerufen werden:

```
#!/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()
```

Beispiel für Hooks

Hier finden Sie ein Beispiel für einen hooks-Eintrag, der zwei Hooks für das `AfterInstall`-Lebenszyklusereignis angibt:

```
hooks:
   AfterInstall:
     - location: Scripts/RunResourceTests.sh
       timeout: 180
     - location: Scripts/PostDeploy.sh
       timeout: 180
```

Das Skript `Scripts/RunResourceTests.sh` wird während der Stufe `AfterInstall` des Bereitstellungsvorgangs ausgeführt. Die Implementierung schlägt fehl, wenn die Ausführung des Skripts mehr als 180 Sekunden (3 Minuten) dauert.

Der Speicherort von Skripts, den Sie im Abschnitt „Hooks“ angeben, ist relativ zum Stamm des Anwendungsrevisionspakets. Im obigen Beispiel befindet sich eine Datei mit dem Namen `RunResourceTests.sh` in einem Verzeichnis mit dem Namen `Scripts`. Das Verzeichnis `Scripts` befindet sich auf der Stammebene des Pakets. Weitere Informationen finden Sie unter Planen Sie eine Überarbeitung für CodeDeploy.

AppSpec Beispiel für eine Datei

Dieses Thema enthält AppSpec Beispieldateien für ein AWS Lambda und eine EC2/On-Premises Bereitstellung.

Themen

- AppSpec Dateibeispiel für eine Amazon ECS-Bereitstellung
- AppSpec Dateibeispiel für ein AWS Lambda-Bereitstellung
- AppSpec Beispiel für eine Datei für eine Bereitstellung EC2/On-Premises

AppSpec Dateibeispiel für eine Amazon ECS-Bereitstellung

Hier ist ein Beispiel für eine in YAML geschriebene AppSpec Datei für die Bereitstellung eines Amazon ECS-Service.

```
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"
```

Hier finden Sie eine Version des obigen Beispiels in 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"
        }
    ]
}
```

Hier finden Sie die Ereignisabfolge während der Bereitstellung:

1. Bevor die aktualisierte Amazon ECS-Anwendung auf dem Ersatz-Tasksatz installiert wird, wird die Lambda-Funktion mit dem Namen `LambdaFunctionToValidateBeforeInstall` ausgeführt.
2. Nachdem die aktualisierte Amazon ECS-Anwendung auf dem Ersatzaufgabensatz installiert wurde, aber bevor sie Datenverkehr empfängt, wird die Lambda-Funktion namens Lambda-Funktion `LambdaFunctionToValidateAfterInstall` ausgeführt.
3. Nachdem die Amazon ECS-Anwendung auf dem Ersatzaufgabensatz anfängt, Datenverkehr vom Test-Listener zu empfangen, wird die Lambda-Funktion namens `LambdaFunctionToValidateAfterTestTrafficStarts` Lambda-Funktion ausgeführt. Diese Funktion führt wahrscheinlich Validierungstests durch, um zu bestimmen, ob die Bereitstellung fortgesetzt wird. Wenn Sie keinen Test-Listener in Ihrer Bereitstellungsgruppe angeben, wird dieser Hook ignoriert.
4. Nachdem alle Validierungstests im `AfterAllowTestTraffic` Hook abgeschlossen sind und bevor der Produktionsdatenverkehr an die aktualisierte Amazon ECS-Anwendung weitergeleitet wird, wird die Lambda-Funktion mit dem Namen `LambdaFunctionToValidateBeforeAllowingProductionTraffic` ausgeführt.
5. Nachdem der Produktionsdatenverkehr für die aktualisierte Amazon ECS-Anwendung auf dem Ersatz-Tasksatz bereitgestellt wurde, wird die Lambda-Funktion namens Lambda-Funktion `LambdaFunctionToValidateAfterAllowingProductionTraffic` ausgeführt.

Die Lambda-Funktionen, die während eines beliebigen Hooks ausgeführt werden, können Validierungstests durchführen oder Verkehrsmetriken sammeln.

AppSpec Dateibeispiel für ein AWS Lambda-Bereitstellung

Hier ist ein Beispiel für eine in YAML geschriebene AppSpec Datei für die Bereitstellung einer Lambda-Funktionsversion.

```
version: 0.0
Resources:
  - myLambdaFunction:
      Type: AWS::Lambda::Function
      Properties:
        Name: "myLambdaFunction"
        Alias: "myLambdaFunctionAlias"
        CurrentVersion: "1"
        TargetVersion: "2"
Hooks:
  - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeTrafficShift"
  - AfterAllowTraffic: "LambdaFunctionToValidateAfterTrafficShift"
```

Hier finden Sie eine Version des obigen Beispiels in JSON.

```
{
  "version": 0.0,
  "Resources": [{
   "myLambdaFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
     "Name": "myLambdaFunction",
     "Alias": "myLambdaFunctionAlias",
     "CurrentVersion": "1",
     "TargetVersion": "2"
    }
   }
  }],
  "Hooks": [{
    "BeforeAllowTraffic": "LambdaFunctionToValidateBeforeTrafficShift"
      },
      {
    "AfterAllowTraffic": "LambdaFunctionToValidateAfterTrafficShift"
   }
  ]
 }
```

Hier finden Sie die Ereignisabfolge während der Bereitstellung:

1. Bevor Sie den Datenverkehr von Version 1 einer aufgerufenen Lambda-Funktion `myLambdaFunction` auf Version 2 verlagern, führen Sie eine Lambda-Funktion namens aus, `LambdaFunctionToValidateBeforeTrafficShift` die bestätigt, dass die Bereitstellung bereit ist, mit der Verkehrsverlagerung zu beginnen.
2. Wenn `LambdaFunctionToValidateBeforeTrafficShift` den Exit-Code 0 (Erfolg) zurückgibt, fangen Sie an, Datenverkehr in Version 2 von `myLambdaFunction` zu verschieben. Die Bereitstellungskonfiguration für diese Bereitstellung bestimmt die Geschwindigkeit, in der der Datenverkehr verschoben wird.
3. Nachdem die Übertragung des Datenverkehrs von Version 1 einer aufgerufenen Lambda-Funktion `myLambdaFunction` auf Version 2 abgeschlossen ist, führen Sie eine aufgerufene Lambda-Funktion aus, `LambdaFunctionToValidateAfterTrafficShift` die bestätigt, dass die Bereitstellung erfolgreich abgeschlossen wurde.

AppSpec Beispiel für eine Datei für eine Bereitstellung EC2/On-Premises

Hier ist ein Beispiel für eine AppSpec Datei für eine direkte Bereitstellung auf einer Amazon Linux-, Ubuntu Server- oder RHEL-Instance.

> **Note**
>
> Bereitstellungen auf Windows Server-Instances unterstützen das Element nicht. `runas` Wenn Sie es auf Windows Server-Instanzen bereitstellen, nehmen Sie es nicht in Ihre AppSpec Datei auf.

```
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
```

Wechseln Sie für eine Windows Server-Instanz `os: linux` zu`os: windows`. Außerdem müssen Sie die `destination`-Pfade vollständig qualifizieren (z. B. `c:\temp\webapps\Config` und `c:\temp\webapps\myApp`). Beziehen Sie das Element `runas` nicht mit ein.

Hier finden Sie die Ereignisabfolge während der Bereitstellung:

1. Führen Sie das Skript unter `Scripts/UnzipResourceBundle.sh` aus.
2. Wenn das vorherige Skript einen Beendigungscode von 0 (Erfolg) wiedergegeben hat, führen Sie das Skript unter `Scripts/UnzipDataBundle.sh` aus.
3. Kopieren Sie den Dateipfad von `Config/config.txt` zum Pfad `/webapps/Config/config.txt`.
4. Kopieren Sie rekursiv alle Dateien im Verzeichnis `source` in das Verzeichnis `/webapps/myApp`.
5. Führen Sie das Skript unter `Scripts/RunResourceTests.sh` mit einem Timeout von 180 Sekunden (3 Minuten) aus.
6. Führen Sie das Skript unter `Scripts/RunFunctionalTests.sh` mit einem Timeout von 3 600 Sekunden (1 Stunde) aus.
7. Führen Sie das Skript unter `Scripts/MonitorService.sh` als Benutzer `codedeploy` mit einem Timeout von 3 600 Sekunden (1 Stunde) aus.

AppSpec Abstand zwischen den Dateien

Im Folgenden finden Sie das richtige Format für den AppSpec Dateiabstand. Die Zahlen in eckigen Klammern geben die Anzahl der Leerzeichen an, die zwischen Elementen eingefügt werden müssen. `[4]`Bedeutet zum Beispiel, vier Leerzeichen zwischen den Elementen einzufügen. CodeDeploy löst einen Fehler aus, der möglicherweise schwer zu debuggen ist, wenn die Speicherorte und die Anzahl der Leerzeichen in einer AppSpec Datei nicht korrekt sind.

```
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
```

Hier ist ein Beispiel für eine Datei mit korrektem Abstand AppSpec :

```
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
```

Weitere Informationen über Leerzeichen finden Sie in der YAML-Spezifikation.

Bestätigen Sie Ihre AppSpec Datei und den Speicherort

Dateisyntax

Sie können das von uns AWS bereitgestellte AppSpec Assistant-Skript verwenden, um den Inhalt einer AppSpec Datei zu überprüfen. Sie finden das Skript zusammen mit den AppSpec Dateivorlagen unter GitHub.

Sie können auch ein browserbasiertes Tool wie YAML Lint oder Online YAML Parser verwenden, um Ihre YAML-Syntax zu überprüfen.

Speicherort der Datei

Führen Sie einen der folgenden Befehle aus, um zu überprüfen, ob Sie Ihre AppSpec Datei im Stammverzeichnis der Verzeichnisstruktur des Quellinhalts der Anwendung abgelegt haben:

Auf lokalen Linux-, macOS- oder Unix-Instances:

```
ls path/to/root/directory/appspec.yml
```

Wenn sich die AppSpec Datei dort nicht befindet, wird die Fehlermeldung „Keine solche Datei oder kein solches Verzeichnis“ angezeigt.

Auf lokalen Windows-Instances:

```
dir path\to\root\directory\appspec.yml
```

Wenn sich die AppSpec Datei dort nicht befindet, wird der Fehler „Datei nicht gefunden“ angezeigt.

CodeDeploy Referenz zur Agentenkonfiguration

Wenn der CodeDeploy Agent installiert ist, wird eine Konfigurationsdatei auf der Instanz platziert. Diese Konfigurationsdatei gibt Verzeichnispfade und andere Einstellungen an CodeDeploy , die bei der Interaktion mit der Instanz verwendet werden sollen. Sie können einige der Konfigurationsoptionen in der Datei ändern.

Für Amazon Linux-, Ubuntu Server- und Red Hat Enterprise Linux (RHEL) -Instances hat die Konfigurationsdatei einen Namen`codedeployagent.yml`. Sie befindet sich im Verzeichnis `/etc/codedeploy-agent/conf`.

Für Windows Server-Instances hat die Konfigurationsdatei einen Namen`conf.yml`. Sie befindet sich im Verzeichnis `C:\ProgramData\Amazon\CodeDeploy`.

Die Konfigurationseinstellungen sind:

:log_aws_wire:	Ist auf eingestellt, **`true`** damit der CodeDeploy Agent Wire-Logs von Amazon S3 erfasst und in eine Protokolldatei an dem Speicherort schreibt, auf den die Einstellung:log_dir: verweist. Für Version 2.0.x und höher lautet der Dateiname `codedeploy-agent.aws_wire.log` (erstellt mit 0640-Berechtigungen). Für Version 1.8.x und früher lautet der Dateiname. `codedeploy-agent.wire.log` **Warning** Setzen Sie für : log_aws_wire: die Einstellung nur für die Dauer der Wire-Protokollerfassung auf `true`. Die Wire-Log-Datei kann schnell sehr groß werden. Die Wire-Log-Ausgabe kann

	vertrauliche Informationen enthalten, darunter Klartext-Inhalte von Dateien, die in Amazon S3 oder aus Amazon S3 übertragen wurden, während diese Einstellung auf gesetzt war. `true` Die Wire-Logs enthalten Informationen über alle Amazon S3 S3-Aktivitäten, die mit dem AWS Konto verknüpft waren, während diese Einstellung aktiviert war`true`, nicht nur über Aktivitäten im Zusammenhang mit CodeDeploy Bereitstellungen. Die Standardeinstellung lautet `false`. Diese Einstellung gilt für alle Instance-Typen. Sie müssen diese Konfigurationseinstellung zu Windows Server-Instances hinzufügen, um sie verwenden zu können.
:log_dir:	Der Ordner auf der Instanz, in dem Protokoll dateien zu CodeDeploy Agentenvorgängen gespeichert werden. Die Standardeinstellung gilt `'/var/log/aws/codedeploy-agent'` für Amazon Linux-, Ubuntu Server- und RHEL-Instances sowie `C:\ProgramData\Amazon\CodeDeploy\log` für Windows Server-Instances.

:pid_dir:	Der Ordner, in dem `codedeploy-agent.pid` gespeichert ist. Diese Datei enthält die Prozess-ID (PID) des CodeDeploy Agenten. Die Standardeinstellung lautet `'/opt/codedeploy-agent/state/.pid'` . Diese Einstellung gilt nur für Amazon Linux-, Ubuntu Server- und RHEL-Instances.
:program_name:	Der Name des CodeDeploy Agent-Programms. Die Standardeinstellung lautet `codedeploy-agent`. Diese Einstellung gilt nur für Amazon Linux-, Ubuntu Server- und RHEL-Instances.
:root_dir:	Der Ordner, in dem verwandte Revisionen, Bereitstellungsverlauf und Bereitstellungsscripts auf der Instance gespeichert werden. Die Standardeinstellung gilt `/opt/codedeploy-agent/deployment-root` für Amazon Linux-, Ubuntu Server- und RHEL-Instances sowie `C:\ProgramData\Amazon\CodeDeploy` für Windows Server-Instances.
:verbose:	Ist auf eingestellt, `true` damit der CodeDeploy Agent Debug-Meldungen und Protokolldateien auf der Instance ausgibt. Die Standardeinstellung lautet `false`.

:wait_between_runs:	Das Intervall in Sekunden zwischen der Abfrage ausstehender Bereitstellungen durch CodeDeploy den CodeDeploy Agenten. Für Version 2.0.x und höher ist die Standarde instellung. `30` Für Version 1.8.x und früher ist die Standarde instellung. `1`
:on_premises_config_file:	Für lokale Instanzen der Pfad zu einem alternativen Speicherort für die Konfigura tionsdatei mit dem Namen `codedeplo y.onpremises.yml` (für Ubuntu Server und RHEL) oder `conf.onpremises.yml` (für Windows Server). Standardmäßig werden diese Dateien in`/etc/ codedeploy-agent/conf /codedeplo y.onpremises.yml` für Ubuntu Server und RHEL und in`C:\ProgramData\Amazon \CodeDeploy \conf.onpremises.yml` für Windows Server gespeichert. In Version 1.0.1.686 und höheren Versionen des CodeDeploy-Agenten verfügbar.
:proxy_uri:	(Optional) Der HTTP-Proxy, über den der CodeDeploy Agent AWS für Ihre CodeDeplo y Operationen eine Verbindung herstellen soll. Verwenden Sie ein Format wie dieses: `https://user:password@my.pr oxy:443/path?query` . In Version 1.0.1.824 und höheren Versionen des CodeDeploy-Agenten verfügbar.

:max_revisions:	(Optional) Die Anzahl der Anwendungsrevisionen für eine Bereitstellungsgruppe, die der CodeDeploy Agent archivieren soll. Revisionen, die die angegebene Zahl überschreiten, werden gelöscht. Geben Sie eine positive Ganzzahl ein. Wenn kein Wert angegeben ist, CodeDeploy werden zusätzlich zur aktuell bereitgestellten Version die fünf neuesten Versionen beibehalten. Unterstützt in Version 1.0.1.966 und in höheren Versionen des CodeDeploy-Agenten.
Wenn kein Wert angegeben ist, werden die fünf neuesten Versionen zusätzlich zur aktuell bereitgestellten Version beibehalten. ----sep----:enable_auth_policy:	(Optional) Stellen Sie diese Option ein, `true` wenn Sie die IAM-Autorisierung verwenden möchten, um die Zugriffskontrolle zu konfigurieren und die Berechtigungen der IAM-Rolle oder des Benutzers, die der Agent verwendet, einzuschränken. CodeDeploy ZuVerwendung CodeDeploy mit Amazon Virtual Private Cloud, dieser Wert muss sein. `true` Die Standardeinstellung lautet `false`.
Für muss dieser Wert sein. ----sep----:disable_imds_v1:	Diese Einstellung ist mit Agent 1.7.0 und höher verfügbar. CodeDeploy Wird auf gesetzt, `true` um den Fallback auf IMDSv1 zu deaktivieren, wenn IMDSv2-Fehler auftreten. Die Standardeinstellung ist (aktiviert den Fallback). `false`

Standardmäßig (aktiviert den Fallback). ----sep----:disable_core_dumps:	Für Version 2.0.x und höher. Auf einstellen, `true` um Core-Dumps zu unterdrücken. Auf `false` für die Absturzun tersuchung eingestellt. Die Standardeinstellung lautet `true`.
Auf für die Absturzuntersuchung gesetzt. ----sep----:enable_command_port:	Für Version 2.0.x und höher. Wird auf gesetzt, `true` um die lokale Verwaltun gsschnittstelle (127.0.0.1) nur für Loopback zu aktivieren, um den Agentenstatus abzufragen und Befehle lokal einzufügen. Die Standardeinstellung lautet `false`.
Wird auf gesetzt, um die lokale Verwaltun gsschnittstelle nur für Loopback (127.0.0.1) zum Abfragen des Agentenstatus und zum lokalen Einfügen von Befehlen zu aktivieren. ----sep----:enable_archive_reuse:	Für Version 2.1.0 und höher. Stellen Sie diese Option ein, `false` um die Anwendungsversion immer herunterzuladen, auch in einer `RESTART` Bereitstellung, bei der sich die Revision bereits auf der Instanz befindet. Wenn diese Einstellung aktiviert ist`true`, installiert der Agent von der Kopie, die sich bereits auf der Instanz befindet, und lädt die Revision herunter, falls diese Kopie fehlt oder unvollständig ist. Die Standardeinstellung lautet `true`. Weitere Informationen über dieses Verhalten finden Sie unter Versionsverlauf des Agenten CodeDeploy.

Wenn diese Einstellung aktiviert ist, installiert der Agent von der Kopie, die sich bereits auf der Instanz befindet, und lädt die Revision herunter, falls diese Kopie fehlt oder unvollständig ist. ----sep----:reject_symlinks_in_bundle:	Für Version 2.0.x und höher. Auf einstellen, `true` um Bundles abzulehnen, die Symlinks oder Hardlinks enthalten. Die Standardeinstellung lautet `false`.
Wird auf gesetzt, um Bundles abzulehnen, die Symlinks oder Hardlinks enthalten. ----sep----:reject_path_traversal_in_bundle:	Für Version 2.0.x und höher. Auf festlegen, `true` dass Archiveinträge, die `../` Pfaddurchquerung oder absolute Pfade enthalten, sowie AppSpec Quellfelder, die dem Bundle-Stammverzeichnis entgehen, zurückgewiesen werden. Die Standardeinstellung lautet `false`.
Wird auf gesetzt, um Archiveinträge zurückzuweisen, die Pfaddurchquerung oder absolute Pfade enthalten, sowie Quellfelder, die dem Paketstamm entgehen. ----sep----:reject_unsafe_permissions_in_bundle:	Für Version 2.0.x und höher. Auf einstellen, `true` um Bundles abzulehnen, die SUID/SGID Dateien oder festgelegte AppSpec Berechtigungsmodi enthalten. SUID/SGID Die Standardeinstellung lautet `false`.
Auf setzen, um Pakete abzulehnen, die Dateien enthalten, oder Berechtigungsmodi, die festgelegt sind. ----sep----:reject_unconfined_selinux_in_bundle:	Für Version 2.0.x und höher. Auf einstellen, `true` um Bundles abzulehnen, die Dateien mit MAC-disabling SELinux-Typen enthalten. Die Standardeinstellung lautet `false`.

Wird auf gesetzt, um Bundles abzulehnen, die Dateien mit SELinux-Typen enthalten. ----sep----:reject_symlink_permission_targets:	Für Version 2.0.x und höher. Wird auf gesetzt`true`, um einen AppSpec `permissions` Dateieintrag zurückzuw eisen, dessen Ziel ein symbolischer Link ist. Standardmäßig folgt der Agent dem Link, wenn er Eigentums-, Modus-, ACL- und SELinux-Ä nderungen vornimmt. Die Standardeinstellung lautet `false`.
Standardmäßig folgt der Agent dem Link, wenn er Eigentums-, Modus-, ACL- und SELinux-Änderungen anwendet. ----sep----:ignore _ownership_in_bundle:	Für Version 2.0.x und höher. Stellen Sie diese `true` Option ein, um den Besitz von Dateien, aus denen extrahiert wurde, `tar` und `tgz` Bundles auf zurückzus etzen, `root` anstatt die im Archiv aufgezeic hnete Benutzer-ID und Gruppen-ID zu übernehmen. Dateien in `zip` Paketen gehören `root` unabhängig von dieser Einstellung dem Eigentümer. Die Standardeinstellung lautet `false`.
Dateien in Bundles gehören unabhängig von dieser Einstellung dem Eigentümer. ----sep----:restrict_agent_dir_permissions:	Für Version 2.0.x und höher. Nur Linux-Ins tanzen. Stellen Sie diese Option ein`true`, um die Verzeichnisse und Dateien im Stammverz eichnis der Agenteninstallation auf den reinen Root-Zugriff zu beschränken und nicht auf die standardmäßig verwendeten Modi, die für alle lesbar sind. Die Standardeinstellung lautet `false`.

Wird auf gesetzt, um die Verzeichnisse und Dateien im Stammverzeichnis der Agenteninstallation auf reinen Root-Zugriff zu beschränken und nicht auf die standardmäßig für alle lesbaren Modi. ----sep----:restrict_log_dir_permissions:	Für Version 2.0.x und höher. Nur Linux-Installanzen. Wird auf gesetzt, `true` um das Agent-Log-Verzeichnis, das das Agent-Log und das Updater-Log enthält, auf reinen Root-Zugriff statt auf die standardmäßig verwendeten, allgemein lesbaren Modi zu beschränken. Die Standardeinstellung lautet `false`.
Wird auf gesetzt, um das Agent-Log-Verzeichnis, das das Agent-Log und das Updater-Log enthält, auf reinen Root-Zugriff statt auf die standardmäßig für alle lesbaren Modi zu beschränken. ----sep----:restrict_hook_env_to_allowlist:	Für Version 2.0.x und höher. Auf einstellen, `true` um Lifecycle-Event-Hooks mit einer minimalen Umgebung auszuführen, die nur grundlegende Shell-Variablen und die CodeDeploy Deployment-Variablen enthält. Standardmäßig erben Hooks die gesamte Umgebung des Agentenprozesses. Die Standardeinstellung lautet `false`.

Standardmäßig erben Hooks die gesamte Umgebung des Agentenprozesses. ----sep----:strip_loader_env_in_hooks:	Für Version 2.0.x und höher. Linux-Instanzen und macOS-Instanzen mit Version 2.1.2 und höher. Auf einstellen, `true` um die `LD_AUDIT` Variablen `LD_PRELOAD` `LD_LIBRARY_PATH` , und aus der Umgebung von Lifecycle-Event-Hooks zu entfernen. Unter macOS entfernt die Einstellung stattdessen die `DYLD_VERSIONED_FRAMEWORK_PATH` Variablen `DYLD_INSERT_LIBRARIES` `DYLD_LIBRARY_PATH` `DYLD_FRAMEWORK_PATH` `DYLD_FALLBACK_LIBRARY_PATH` , `DYLD_FALLBACK_FRAMEWORK_PATH` `DYLD_VERSIONED_LIBRARY_PATH` ,,, und. Die Standardeinstellung lautet `false`.
Unter macOS entfernt die Einstellung stattdessen die Variablen,,,, und. ----sep----:disable_powershell_profile_in_hooks:	Für Version 2.0.x und höher. Nur Windows Server-Instanzen. Auf einstellen, `true` um `.ps1` Lifecycle-Event-Hooks mit den `-NonInteractive` Optionen `-NoProfile` und auszuführen. Standardmäßig laden diese Hooks das PowerShell Profil des Benutzers, unter dem der Agent ausgeführt wird. Die Standardeinstellung lautet `false`.

Standardmäßig laden diese Hooks das Profil des Benutzers, unter dem der Agent läuft. ----sep----:archive_max_extraction_size:	Für Version 2.0.x und höher. Legt eine Obergrenze für die deklarierte unkomprimierte Bundle-Größe fest. Wenn der extrahierte Inhalt diesen Wert überschreitet, schlägt die Bereitstellung fehl. Die Standardeinstellung ist keine Obergrenze (unbegrenzt).
Die Standardeinstellung ist keine Obergrenze (unbegrenzt). ----sep----:wait_after_error:	Die Anzahl der Sekunden, die der Agent nach dem Auftreten eines Fehlers wartet, bevor er den normalen Betrieb wieder aufnimmt. Die Standardeinstellung lautet `30`.
Die Anzahl der Sekunden, die der Agent nach einem Fehler wartet, bevor er den normalen Betrieb wieder aufnimmt. ----sep----:http_r ead_timeout:	Das HTTP-Lese-Timeout in Sekunden für Service-API-Aufrufe. Die Standardeinstellung lautet `80`.
Das HTTP-Lese-Timeout in Sekunden für Service-API-Aufrufe. ----sep----:kill_agent_max_ wait_time_seconds:	Die maximale Anzahl von Sekunden, die der Agent auf ein ordnungsgemäßes Herunterf ahren wartet, bevor er den Vorgang erzwingt. Für Version 2.0.x und höher ist dieser Wert auf 24 Stunden (86400 Sekunden) begrenzt. Die Standardeinstellung ist `7200` (2 Stunden).
Die Standardeinstellung ist (2 Stunden). ----sep----:enable_deployments_log:	Auf setzen, um das aggregierte `true` Bereitste llungsprotokoll () zu schreiben. `codedeploy-agent-deployments.log` Die Standardeinstellung lautet `true`.
Legt fest, dass das aggregierte Bereitste llungsprotokoll () geschrieben werden soll. ----sep----:use_fips_mode:	Auf einstellen, um Endpunkte zu verwenden. `true` FIPS-compliant Gilt nur für US-Regionen. Die Standardeinstellung lautet `false`.

Gilt nur für US-Regionen. ----sep----:deploy _control_endpoint:	(Optional) Eine benutzerdefinierte URL für den Steuerungsebenen-Endpunkt für den Service. CodeDeploy Die Standardeinstellung ist kein Überschreiben (der Agent löst den Endpunkt aus der Region auf).
Die Standardeinstellung ist kein Override (der Agent löst den Endpunkt aus der Region auf). ----sep----:s3_endpoint_override:	(Optional) Eine benutzerdefinierte Amazon S3 S3-Endpunkt-URL für Bundle-Downloads. Die Standardeinstellung ist kein Überschreiben (der Agent löst den Amazon S3 S3-Endpunkt aus der Region auf).
Die Standardeinstellung ist kein Überschreiben (der Agent löst den Amazon S3-Endpunkt aus der Region auf). ----sep----:s3_download_max _concurrency:	Für Version 2.1.2 und höher. Die Anzahl der parallel HTTP-Bereichsanfra gen, die der Agent verwendet, um eine Anwendungsversion von Amazon S3 herunterz uladen. Auf einen Wert setzen, der größer als ist1, um große Versionen in parallel Bereichen herunterzuladen. Mit `1` lädt der Agent die Revision mit einer einzigen `GetObject` Anfrage herunter. `64`Die obigen Werte werden als `64` und `0` werden als behandelt1. Die Standardeinstellung lautet `1`.

Die obigen Werte werden als und werden behandelt als. ----sep----:s3_download_chunk_size:	Für Version 2.1.2 und höher. Die Größe jeder Bereichsanforderung, wenn sie größer als `:s3_download_max_concurrency: 1` ist. Geben Sie eine Bytezahl oder eine Größenzeichenfolge wie `"16MiB"` oder ein`"16MB"`. Eine Version dieser Größe oder kleiner wird mit einer Anfrage heruntergeladen. Werte unter 64 KiB werden als 64 KiB behandelt. Die Standardeinstellung lautet `"8MiB"`.
Werte unter 64 KiB werden als 64 KiB behandelt. ----sep----:ongoing_deployment_tracking:	Der Name des Unterverzeichnisses, das für die Statusverfolgung der Bereitstellung verwendet wird. Die Standardeinstellung lautet `ongoing-deployment` .

Verwandte Themen

Mit dem CodeDeploy Agenten arbeiten

Verwaltung des CodeDeploy Agentenbetriebs

CloudFormation Vorlagen als CodeDeploy Referenz

In diesem Abschnitt werden die CloudFormation Ressourcen, Transformationen und Hooks vorgestellt, die für die Verwendung mit CodeDeploy Bereitstellungen konzipiert wurden. Eine exemplarische Vorgehensweise zur Erstellung eines Stack-Updates, das vom CloudFormation Hook für CodeDeploy verwaltet wird, finden Sie unter Erstellen Sie eine Amazon blue/green ECS-Bereitstellung über CloudFormation

> **Note**
>
> CloudFormation Hooks sind Teil der CloudFormation Komponenten für CodeDeploy Lifecycle-Events AWS und unterscheiden sich von diesen.

Zusätzlich zu den anderen Methoden, die Ihnen in zur Verfügung stehen CodeDeploy, können Sie CloudFormation Vorlagen verwenden, um die folgenden Aufgaben auszuführen:

- Erstellen von Anwendungen
- Erstellen von Bereitstellungsgruppen und Angeben einer Zielrevision
- Erstellen von Bereitstellungskonfigurationen
- Erstellen Sie Amazon EC2 EC2-Instances.

CloudFormation ist ein Service, der Ihnen hilft, Ihre AWS Ressourcen mithilfe von Vorlagen zu modellieren und einzurichten. Eine CloudFormation Vorlage ist eine Textdatei, deren Format dem JSON-Standard entspricht. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen beschreibt, und CloudFormation kümmert sich um die Bereitstellung und Konfiguration dieser Ressourcen für Sie.

Weitere Informationen finden Sie unter Was ist AWS CloudFormation? und Arbeiten mit AWS CloudFormation Vorlagen im AWS CloudFormation Benutzerhandbuch.

Wenn Sie beabsichtigen, CloudFormation Vorlagen zu verwenden, die CodeDeploy in Ihrer Organisation kompatibel sind, müssen Sie als Administrator Zugriff auf die AWS Dienste CloudFormation und Aktionen gewähren, CloudFormation von denen dies abhängt. Um Berechtigungen zum Erstellen von Anwendungen, Bereitstellungsgruppen und Bereitstellungskonfigurationen zu erteilen, fügen Sie dem Berechtigungssatz der Benutzer, mit denen Sie arbeiten, die folgende Richtlinie hinzu CloudFormation:

JSON

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

```
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Weitere Informationen zu Richtlinien finden Sie in den folgenden Themen:

- Informationen zu den Richtlinien, die zum Berechtigungssatz der Benutzer hinzugefügt werden müssen, die Amazon EC2 EC2-Instances erstellen, finden Sie unterErstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy (CloudFormation Vorlage).
- Informationen zum Hinzufügen von Richtlinien zu Berechtigungssätzen finden Sie unter Erstellen eines Berechtigungssatzes im IAM-Benutzerhandbuch.
- Informationen zum Beschränken von Benutzern auf eine begrenzte Anzahl von CodeDeploy Aktionen und Ressourcen finden Sie unterAWS verwaltete (vordefinierte) Richtlinien für CodeDeploy.

Die folgende Tabelle zeigt die Aktionen, die eine CloudFormation Vorlage in Ihrem Namen ausführen kann, und enthält Links zu weiteren Informationen zu den AWS Ressourcentypen und ihren Eigenschaftstypen, die Sie einer CloudFormation Vorlage hinzufügen können.

Action	CloudFormation Referenz	Art der Referenz
Erstellen Sie eine CodeDeploy Anwendung.	AWS::Code Deploy: :Anwendung	CloudFormation Ressource
Erstellen und spezifizieren Sie die Details für eine Bereitstellungsgruppe, die für die Bereitstellung Ihrer Anwendungsrevisionen verwendet werden soll. [1]	AWS::CodeDeploy::D eploymentGroup	CloudFormation Ressource
Erstellen Sie eine Reihe von Bereitstellungsreg eln, Bedingungen für den	AWS::CodeDeploy::D eploymentConfig	CloudFormation Ressource

Action	CloudFormation Referenz	Art der Referenz
Erfolg der Bereitstellung und Bedingungen für Fehlschläge bei der Bereitstellung, die während einer Bereitstellung verwendet CodeDeploy werden.		
Erstellen Sie eine Amazon EC2 EC2-Instance. [2]	AWS: :EC2: :Instanz	CloudFormation Ressource
Verwenden Sie CloudFormation `AWS::CodeDeployBlueGreen` Transform und `AWS::CodeDeploy::BlueGreen` Hook, um Stack-Updates zu verwalten, Ressourcen zu erstellen und den Datenverkehr für CodeDeploy blue/green Bereitstellungen zu verlagern. [3]	AWS::CodeDeployBlueGreen	Die `AWS::CodeDeployBlueGreen` -Transformation ist ein Makro, das von CloudFormation gehostet wird.
	AWS::CodeDeploy::BlueGreen	Der `AWS::CodeDeploy::BlueGreen` Hook ist als Hook Ressource in strukturiert CloudFormation. Der Hook enthält Parameter, die an die Stelle Ihrer CodeDeploy AppSpec Datei treten, indem sie auf bestimmte CodeDeploy Lifecycle-Event-Hooks verweisen.

Action	CloudFormation Referenz	Art der Referenz
[1] Wenn Sie die Version der Anwendungsrevision angeben, die Sie als Teil der Bereitstellungsgruppe bereitstellen möchten, wird Ihre Zielrevision bereitgestellt, sobald der Bereitstellungsprozess abgeschlossen ist. Weitere Informationen zur Vorlagenkonfiguration finden Sie unter CodeDeploy DeploymentGroup Bereitstellungsrevision S3Location und CodeDeploy DeploymentGroup Deployment-Revision GitHubLocation im AWS CloudFormation Benutzerhandbuch. [2] Wir stellen Vorlagen zur Verfügung, mit denen Sie Amazon EC2 EC2-Instances in den Regionen erstellen können, in denen dies unterstützt CodeDeploy wird. Weitere Informationen zum Verwenden dieser Vorlagen finden Sie unter Erstellen Sie eine Amazon EC2 EC2-Instance für CodeDeploy (CloudFormation Vorlage). [3] Nur Amazon blue/green ECS-Bereitstellungen werden von dieser Bereitstellungskonfiguration unterstützt. Weitere Informationen zu Bereitstellungskonfigurationen für Amazon blue/green ECS-Bereitstellungen bis finden Sie CloudFormation unterBereitstellungskonfigurationen für CloudFormation blue/green Bereitstellungen (Amazon ECS). Weitere Informationen zu Amazon blue/green ECS-Bereitstellungen CloudFormation und zur Anzeige Ihrer Bereitstellung finden Sie CodeDeploy unterErstellen Sie eine Amazon blue/green ECS-Bereitstellung über CloudFormation.		

Verwendung CodeDeploy mit Amazon Virtual Private Cloud

Wenn Sie Amazon Virtual Private Cloud (Amazon VPC) zum Hosten Ihrer AWS Ressourcen verwenden, können Sie eine private Verbindung zwischen Ihrer VPC und herstellen. CodeDeploy Sie können diese Verbindung verwenden, CodeDeploy um mit Ihren Ressourcen auf Ihrer VPC zu kommunizieren, ohne das öffentliche Internet nutzen zu müssen.

Amazon VPC ist ein AWS Service, mit dem Sie AWS Ressourcen in einem von Ihnen definierten virtuellen Netzwerk starten können. Mit einer VPC haben Sie die Kontrolle über Ihre Netzwerkeinstellungen, wie IP-Adressbereich, Subnetze, Routing-Tabellen und Netzwerk-Gateways. Bei VPC-Endpunkten wird das Routing zwischen der VPC und den AWS Diensten vom AWS Netzwerk abgewickelt, und Sie können IAM-Richtlinien verwenden, um den Zugriff auf Dienstressourcen zu steuern.

Um Ihre VPC zu verbinden CodeDeploy, definieren Sie einen VPC-Schnittstellen-Endpunkt für. CodeDeploy Ein Schnittstellenendpunkt ist eine elastic network interface mit einer privaten IP-Adresse, die als Einstiegspunkt für Datenverkehr dient, der für einen unterstützten AWS Dienst bestimmt ist. Der Endpunkt bietet zuverlässige, skalierbare Konnektivität, CodeDeploy ohne dass ein Internet-Gateway, eine NAT-Instanz (Network Address Translation) oder eine VPN-Verbindung erforderlich sind. Weitere Informationen finden Sie unter Was ist Amazon VPC im Benutzerhandbuch zu Amazon VPC.

Schnittstelle, auf der VPC-Endpunkte basieren AWS PrivateLink, eine AWS Technologie, die private Kommunikation zwischen AWS Diensten über eine elastic network interface mit privaten IP-Adressen ermöglicht. Weitere Informationen finden Sie unter AWS PrivateLink.

Die folgenden Schritte sind für Benutzer von Amazon VPC vorgesehen. Weitere Informationen finden Sie unter Erste Schritte im Amazon VPC Benutzerhandbuch.

Verfügbarkeit

CodeDeploy hat zwei VPC-Endpunkte: einen für CodeDeploy Agentenoperationen und einen für CodeDeploy API-Operationen. Die folgende Tabelle zeigt die unterstützten AWS Regionen für jeden Endpunkt.

Weitere Informationen und Informationen zu FIPS-Endpunkten finden Sie unter AWS CodeDeploy Endpunkte und Kontingente.

Name der Region	Regionscode	Agenten-Endpunkt	API-Endpunkt	FIPS-konforme Region?
USA Ost (Nord-Virginia)	us-east-1	Ja	Ja	Ja
USA Ost (Ohio)	us-east-2	Ja	Ja	Ja

Name der Region	Regionscode	Agenten-Endpunkt	API-Endpunkt	FIPS-konforme Region?
USA West (Nordkalifornien)	us-west-1	Ja	Ja	Ja
USA West (Oregon)	us-west-2	Ja	Ja	Ja
Afrika (Kapstadt)	af-south-1	Ja	Ja	
Asien-Pazifik (Hongkong)	ap-east-1	Ja	Ja	
Asien-Pazifik (Hyderabad)	ap-south-2	Ja	Ja	
Asien-Pazifik (Jakarta)	ap-southeast-3	Ja	Ja	
Asien-Pazifik (Melbourne)	ap-southeast-4	Ja	Ja	
Asien-Pazifik (Malaysia)	ap-southeast-5	Ja	Ja	
Asien-Pazifik (Neuseeland)	ap-southeast-6	Ja	Ja	
Asien-Pazifik (Thailand)	ap-southeast-7	Ja	Ja	
Asien-Pazifik (Taipeh)	ap-east-2	Ja	Ja	
Asien-Pazifik (Mumbai)	ap-south-1	Ja	Ja	
Asien-Pazifik (Osaka)	ap-northeast-3	Ja	Ja	

Name der Region	Regionscode	Agenten-Endpunkt	API-Endpunkt	FIPS-konforme Region?
Asien-Pazifik (Seoul)	ap-northeast-2	Ja	Ja	
Asien-Pazifik (Singapore)	ap-southeast-1	Ja	Ja	
Asien-Pazifik (Sydney)	ap-southeast-2	Ja	Ja	
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja	Ja	
Kanada (Zentral)	ca-central-1	Ja	Ja	
China (Peking)	cn-north-1	Ja	Nein	
China (Ningxia)	cn-northwest-1	Nein	Nein	
Europa (Frankfurt)	eu-central-1	Ja	Ja	
Europa (Ireland)	eu-west-1	Ja	Ja	
Europa (London)	eu-west-2	Ja	Ja	
Europa (Milan)	eu-south-1	Ja	Ja	
Europa (Paris)	eu-west-3	Ja	Ja	
Europa (Spain)	eu-south-2	Ja	Ja	
Europa (Stockholm)	eu-north-1	Ja	Ja	
Europa (Zürich)	eu-central-2	Ja	Ja	
Israel (Tel Aviv)	il-central-1	Ja	Ja	

Name der Region	Regionscode	Agenten-Endpunkt	API-Endpunkt	FIPS-konforme Region?
Naher Osten (Bahrain)	me-south-1	Ja	Ja	
Naher Osten (VAE)	me-central-1	Ja	Ja	
Mexiko (Zentral)	mx-central-1	Ja	Ja	
Südamerika (São Paulo)	sa-east-1	Ja	Ja	
AWS GovCloud (US-East)	us-gov-east-1	Nein	Nein	Ja
AWS GovCloud (US-West)	us-gov-west-1	Nein	Nein	Ja

VPC-Endpoints erstellen für CodeDeploy

Um mit der Verwendung CodeDeploy mit Ihrer VPC zu beginnen, erstellen Sie einen VPC-Schnittstellen-Endpunkt für. CodeDeploy CodeDeployerfordert separate Endpunkte für Agenten-Git-Operationen und für CodeDeploy API-Operationen. Je nach Ihren geschäftlichen Anforderungen müssen Sie möglicherweise mehr als einen VPC-Endpunkt erstellen. Wenn Sie einen VPC-Endpunkt für erstellen CodeDeploy, wählen Sie AWS Services und unter Service Name eine der folgenden Optionen aus:

- com.amazonaws. ***region***.codedeploy: Wählen Sie diese Option, wenn Sie einen VPC-Endpunkt für CodeDeploy API-Operationen erstellen möchten. Wählen Sie diese Option beispielsweise, wenn Ihre Benutzer die AWS CLI CodeDeploy API oder die AWS SDKs verwenden, um mit CodeDeploy Vorgängen wie`CreateApplication`, und zu interagieren. `GetDeployment` `ListDeploymentGroups`
- com.amazonaws. ***region***.codedeploy-fips: Wählen Sie diese Option, wenn Sie einen VPC-Endpunkt für CodeDeploy API-Operationen für den FIPS-Endpunkt erstellen möchten.

- com.amazonaws. ***region***.codedeploy-commands-secure: Wählen Sie diese Option, wenn Sie einen VPC-Endpunkt für Agentenoperationen erstellen möchten. CodeDeploy Sie müssen dies auch `true` in Ihrer Agenten-Konfigurationsdatei festlegen und `:enable_auth_policy:` die erforderlichen Berechtigungen anhängen. Weitere Informationen finden Sie unter Konfigurieren Sie die Agenten- und IAM-Berechtigungen CodeDeploy.

Wenn Sie Lambda- oder ECS-Bereitstellungen verwenden, müssen Sie nur einen VPC-Endpunkt für com.amazonaws erstellen. ***region***.codedeploy. Kunden, die Amazon EC2 EC2-Bereitstellungen verwenden, benötigen VPC-Endpunkte für beide com.amazonaws. *region*.codedeploy und com.amazonaws. ***region***.codedeploy-commands-secure.

Konfigurieren Sie die Agenten- und IAM-Berechtigungen CodeDeploy

Um Amazon VPC-Endpoints mit zu verwenden CodeDeploy, müssen Sie `true` in der `:enable_auth_policy:` Agent-Konfigurationsdatei, die sich auf Ihren EC2- oder lokalen Instances befindet, den Wert auf setzen. Weitere Informationen zur Agenten-Konfigurationsdatei finden Sie unter. CodeDeploy Referenz zur Agentenkonfiguration

Außerdem müssen Sie Ihrem Amazon EC2-Instance-Profil (wenn Sie Amazon EC2-Instances verwenden) oder Ihrem IAM-Benutzer oder Ihrer IAM-Rolle (wenn Sie lokale Instances verwenden) die folgenden IAM-Berechtigungen hinzufügen.

```
{
  "Statement": [
    {
      "Action": [
        "codedeploy-commands-secure:GetDeploymentSpecification",
        "codedeploy-commands-secure:PollHostCommand",
        "codedeploy-commands-secure:PutHostCommandAcknowledgement",
        "codedeploy-commands-secure:PutHostCommandComplete"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im Amazon VPC Leitfaden.

CodeDeploy Referenz zum Resource Kit

Viele der Dateien, auf die CodeDeploy ich angewiesen bin, werden in öffentlich zugänglichen AWS Region-specific Amazon S3 S3-Buckets gespeichert. Diese Dateien enthalten Installationsdateien für den CodeDeploy Agenten, Vorlagen und Beispielanwendungsdateien. Wir nennen diese Sammlung von Dateien das CodeDeploy Resource Kit.

Themen

- Bucket-Namen für das Resource Kit nach Regionen
- Inhalt des Resource Kits
- Zeigt eine Liste der Resource Kit-Dateien an
- Laden Sie die Resource Kit-Dateien herunter

Bucket-Namen für das Resource Kit nach Regionen

In dieser Tabelle sind die Namen der *`bucket-name`* Ersatzprodukte aufgeführt, die für einige Verfahren im Handbuch erforderlich sind. Dies sind die Namen der Amazon S3 S3-Buckets, die die CodeDeploy Resource Kit-Dateien enthalten.

> **Note**
>
> Um auf den Amazon S3 S3-Bucket in der Region Asien-Pazifik (Hongkong) zuzugreifen, müssen Sie die Region in Ihrem AWS Konto aktivieren. Weitere Informationen finden Sie unter AWS Regionen verwalten.

Name der Region	***`Bucket-name`*** Ersatz	Region-ID
USA Ost (Nord-Virginia)	aws-codedeploy-us-east-1	us-east-1
USA Ost (Ohio)	aws-codedeploy-us-east-2	us-east-2
USA West (Nordkalifornien)	aws-codedeploy-us-west-1	us-west-1
USA West (Oregon)	aws-codedeploy-us-west-2	us-west-2
Afrika (Kapstadt)	aws-codedeploy-af-south-1	af-south-1

Name der Region	*Bucket-name* Ersatz	Region-ID
Asien-Pazifik (Hongkong)	aws-codedeploy-ap-east-1	ap-east-1
Asien-Pazifik (Hyderabad)	aws-codedeploy-ap-Süd-2	ap-south-2
Asien-Pazifik (Jakarta)	aws-codedeploy-ap-southeast-3	ap-southeast-3
Asien-Pazifik (Melbourne)	aws-codedeploy-ap-southeast-4	ap-southeast-4
Asien-Pazifik (Malaysia)	aws-codedeploy-ap-southeast-5	ap-southeast-5
Asien-Pazifik (Neuseeland)	aws-codedeploy-ap-southeast-6	ap-southeast-6
Asien-Pazifik (Thailand)	aws-codedeploy-ap-southeast-7	ap-southeast-7
Asien-Pazifik (Taipeh)	aws-codedeploy-ap-east-2	ap-east-2
Asien-Pazifik (Mumbai)	aws-codedeploy-ap-south-1	ap-south-1
Asien-Pazifik (Osaka)	aws-codedeploy-ap-northeast-3	ap-northeast-3
Asien-Pazifik (Seoul)	aws-codedeploy-ap-northeast-2	ap-northeast-2
Asien-Pazifik (Singapur)	aws-codedeploy-ap-southeast-1	ap-southeast-1
Asien-Pazifik (Sydney)	aws-codedeploy-ap-southeast-2	ap-southeast-2
Asien-Pazifik (Tokio)	aws-codedeploy-ap-northeast-1	ap-northeast-1

Name der Region	*Bucket-name* Ersatz	Region-ID
Kanada (Zentral)	aws-codedeploy-ca-central-1	ca-central-1
Europa (Frankfurt)	aws-codedeploy-eu-central-1	eu-central-1
Europa (Irland)	aws-codedeploy-eu-west-1	eu-west-1
Europa (London)	aws-codedeploy-eu-west-2	eu-west-2
Europa (Milan)	aws-codedeploy-eu-soud-1	eu-south-1
Europa (Paris)	aws-codedeploy-eu-west-3	eu-west-3
Europa (Spain)	aws-codedeploy-eu-Süd-2	eu-south-2
Europa (Stockholm)	aws-codedeploy-eu-nord-1	eu-north-1
Europa (Zürich)	aws-codedeploy-eu-central-2	eu-central-2
Israel (Tel Aviv)	aws-codedeploy-il-central-1	il-central-1
Middle East (Bahrain)	aws-codedeploy-me-south-1	me-south-1
Naher Osten (VAE)	aws-codedeploy-me-central-1	me-central-1
Mexiko (Zentral)	aws-codedeploy-mx-central-1	mx-central-1
Südamerika (São 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

Inhalt des Resource Kits

In der folgenden Tabelle sind die Dateien im CodeDeploy Resource Kit aufgeführt.

Datei	Description
`LATEST_VERSION`	Eine Datei, die von Aktualisierungsmechanismen wie Amazon EC2 Systems Manager verwendet wird, um die neueste Version des CodeDeploy Agenten zu ermitteln.
`VERSION`	Der automatische Aktualisierungsmechanismus wurde in der CodeDeploy Agentenversion 1.1.0 entfernt und diese Datei wird nicht mehr verwendet. Eine Datei, die von CodeDeploy-Agenten verwendet wird, um sich automatisch zu aktualisieren, wenn sie auf Instances ausgeführt werden.
`codedeploy-agent.noarch.rpm`	Der CodeDeploy Agent für Amazon Linux und Red Hat Enterprise Linux (RHEL). Es kann mehrere Dateien mit demselben Basisdatenamen, aber unterschiedlichen Versionen geben (wie `-1.0-0`).
`codedeploy-agent_all.deb`	Der CodeDeploy Agent für Ubuntu Server. Es kann mehrere Dateien mit demselben Basisdateinamen, aber unterschiedlichen Versionen geben (wie `_1.0-0`).
`codedeploy-agent.msi`	Der CodeDeploy Agent für Windows Server. Es kann mehrere Dateien mit demselben Basisdateinamen, aber unterschiedlichen Versionen geben (wie `-1.0-0`).
`install`	Eine Datei, mit der Sie den CodeDeploy Agenten einfacher installieren können. Das `latestv2/` Präfix dient dem Installationsskript der Version 2.0.x.

Datei	Description
`CodeDeploy_SampleCF_Template.json`	Eine AWS CloudFormation Vorlage, mit der Sie von einer bis drei Amazon EC2 EC2-Instances starten können, auf denen Amazon Linux oder Windows Server ausgeführt wird. Es kann mehrere Dateien mit demselben Basisdateinamen, aber unterschiedlichen Versionen geben (wie `-1.0.0`).
`CodeDeploy_SampleCF_ELB_Integration.json`	Eine AWS CloudFormation Vorlage, mit der Sie eine Beispielwebsite mit Lastenausgleich erstellen können, die auf einem Apache-Webserver ausgeführt wird. Die Anwendung ist so konfiguriert, dass sie sich über alle Availability Zones in der Region erstreckt, in der Sie sie erstellt haben. Diese Vorlage erstellt drei Amazon EC2-Instances und ein IAM-Instance-Profil, um den Instances Zugriff auf die Ressourcen in Amazon S3, Amazon EC2 Auto Scaling und Elastic Load CloudFormation Balancing zu gewähren. Außerdem werden ein Load Balancer und eine Servicerolle erstellt. CodeDeploy
`SampleApp_ELB_Integration.zip`	Eine Beispielanwendungsrevision, die Sie auf einer Amazon EC2 EC2-Instance bereitstellen können, die bei einem Elastic Load Balancing Load Balancer registriert ist.
`SampleApp_Linux.zip`	Eine Beispielanwendungsrevision, die Sie auf einer Amazon EC2 EC2-Instance, auf der Amazon Linux ausgeführt wird, oder auf einer Ubuntu Server- oder RHEL-Instance bereitstellen können. Es kann mehrere Dateien mit demselben Basisdateinamen, aber unterschiedlichen Versionen geben (wie `-1.0`).

Datei	Description
`SampleApp_Windows.zip`	Eine Beispielanwendungsrevision, die Sie auf einer Windows Server-Instance bereitstellen können. Es kann mehrere Dateien mit demselben Basisdateinamen, aber unterschiedlichen Versionen geben (wie `-1.0`).

Zeigt eine Liste der Resource Kit-Dateien an

Um eine Liste von Dateien anzuzeigen, verwenden Sie den aws s3 ls Befehl für Ihre Region.

> **Note**
>
> Die Dateien in jedem Bucket sind so konzipiert, dass sie mit Ressourcen in der entsprechenden Region funktionieren.

- `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`

Laden Sie die Resource Kit-Dateien herunter

Um eine Datei herunterzuladen, verwenden Sie den aws s3 cp Befehl für Ihre Region.

> **Note**
>
> Stellen Sie sicher, dass Sie den Punkt (`.`) fast am Ende nicht vergessen. Damit wird die Datei in das aktuelle Verzeichnis heruntergeladen.

Beispielsweise laden die folgenden Befehle eine einzelne Datei mit der Bezeichnung `SampleApp_Linux.zip` aus einem der `/samples/latest/`-Ordner des Buckets herunter:

- ```
  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
  ```

Verwenden Sie einen der folgenden Befehle für Ihre Region, um alle Dateien herunterzuladen:

- ```
  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 Kontingente

In den folgenden Tabellen werden Kontingente in beschrieben CodeDeploy.

> **Note**
>
> Das EC2/On-Premises Limit für die direkte Bereitstellung in Stunden ist unterschiedlich. Für benutzerdefinierte Bereitstellungskonfigurationen, die vor Juni 2023 erstellt wurden, beträgt das Limit 8 Stunden. Für benutzerdefinierte Bereitstellungskonfigurationen, die im Juni 2023 oder später erstellt wurden, beträgt das Limit 12 Stunden. Für vordefinierte Bereitstellungskonfigurationen beträgt das Limit 12 Stunden.

Name	Standard	Anpas	Description
AWS Lambda-Bereitstellung läuft innerhalb weniger Stunden	Jede unterstützte Region: 50	Nein	Maximale Anzahl von Stunden, die eine AWS Lambda-Bereitstellung ausgeführt werden kann (48 Stunden für die maximale Zeit zwischen der ersten und letzten Verkehrsschicht plus eine Stunde für jeden von zwei möglichen Lifecycle-Hooks)
Zugeordnete Anwendungen pro Konto und Region	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Anwendungen, die einem AWS Konto in

Name	Standard	Anpas	Description
			einer einzelnen Region zugeordnet sind
Zugeordnete Alarme pro Bereitstellungsgruppe	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Alarmen, die einer Bereitstellungsgruppe zugeordnet sind
Auto-Scaling-Gruppen in einer Bereitstellungsgruppe	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl der Amazon-EC2-Auto-Scaling-Gruppen in einer Bereitstellungsgruppe
Gleichzeitige Bereitstellungen pro Konto	Jede unterstützte Region: 1.300	Ja	Maximale Anzahl gleichzeitiger Bereitstellungen, die einem Konto zugeordnet sind. AWS Jede Bereitstellung in einer hochskalierten Amazon EC2 Instance in einer Amazon-EC2-Auto-Scaling-Gruppe zählt als eine einzelne gleichzeitige Bereitstellung für jede Anwendung, mit der die EC2-Instance assoziiert ist.

Name	Standard	Anpas	Description
Gleichzeitige Bereitstellungen pro Bereitstellungsgruppe	Jede unterstützte Region: 1	Nein	Maximale Anzahl gleichzeitiger Bereitstellungen für eine Bereitstellungsgruppe. Dieser Grenzwert verhindert gleichzeitige Bereitstellungen derselben Anwendung in derselben Bereitstellungsgruppe.
Benutzerdefinierte Bereitstellungskonfigurationen pro Konto	Jede unterstützte Region: 50	Nein	Maximale Anzahl von benutzerdefinierten Bereitstellungskonfigurationen, die einem Konto zugeordnet sind AWS
Bereitstellungsgruppen, die einer einzelnen Anwendung zugeordnet sind	Jede unterstützte Region: 1 000	Ja	Maximale Anzahl der Bereitstellungsgruppen in Zusammenhang mit einer einzelnen Anwendung
EC2/On-Premises blue/green Die Bereitstellung wurde in Stunden ausgeführt	Jede unterstützte Region: 109	Nein	Maximale Anzahl von Stunden, die eine EC2/On-Premises blue/green Bereitstellung ausgeführt werden kann (48 Stunden für jedes der beiden oben genannten Grenzwerte plus eine Stunde für jedes der 13 möglichen Lebenszyklusereignisse)

Name	Standard	Anpas	Description
EC2/On-Premises Die Bereitstellung vor Ort erfolgt innerhalb von Stunden	Jede unterstützte Region: 12	Nein	Maximale Anzahl von Stunden, die eine Bereitstellung vor EC2/On-Premises Ort ausgeführt werden kann
Ereignisbenachrichtigungs-Auslöser in einer Bereitstellungsgruppe	Jede unterstützte Region: 10	Yes (Ja)	Maximale Anzahl der Ereignisbenachrich tigungs-Auslöser in einer Bereitstellungsgruppe
GitHub Verbindungstoken pro Konto	Jede unterstützte Region: 25	Nein	Maximale Anzahl von GitHub Verbindungstoken für ein einzelnes AWS Konto
Stunden zwischen dem Abschluss einer Bereitstellung und der Beendigung der ursprünglichen Instanzen während einer EC2/On-Premises blue/green Bereitste llung	Jede unterstützte Region: 48	Nein	Maximale Anzahl von Stunden zwischen dem Abschluss einer Bereitste llung und der Beendigun g der ursprünglichen Instances während einer EC2/On-Premises blue/ green Bereitstellung
Stunden zwischen der Bereitstellung einer Revision und der Verlagerung des Datenverkehrs auf die Ersatzinstanzen während einer EC2/On-Premises blue/ green Bereitstellung	Jede unterstützte Region: 48	Nein	Maximale Anzahl von Stunden zwischen der Bereitstellung einer Revision und der Verlagerung des Datenverkehrs auf die Ersatzinstanzen während einer EC2/On-Premises blue/green Bereitstellung

Name	Standard	Anpas	Description
Anzahl der Instances pro Bereitstellung	us-east-1: 2 000 us-east-2:2.000 us-west-2: 2 000 eu-central-1: 2 000 Jede der anderen unterstützten Regionen: 1 000	Ja	Maximale Anzahl der Instances in einer einzelnen Bereitstellung
Eine blue/green Bereitstellung kann nach einer erfolgreichen Bereitstellung einige Minuten warten, bis Instances aus der ursprünglichen Bereitstellung beendet werden	Jede unterstützte Region: 2 800	Nein	Maximale Anzahl von Minuten, die eine blue/green Bereitstellung nach einer erfolgreichen Bereitstellung warten kann, bevor Instanzen aus der ursprünglichen Bereitstellung beendet werden
Minuten zwischen der ersten und letzten Verkehrsverlagerung während einer AWS Lambda-Bereitstellung (Canary) oder einer linearen Lambda-Bereitstell ung	Jede unterstützte Region: 2 880	Nein	Maximale Anzahl von Minuten zwischen der ersten und der letzten Verkehrsverlagerung während einer AWS Lambda-Bereitstellung (Canary) oder einer linearen Lambda-Be reitstellung

Name	Standard	Anpas	Description
Minuten, bis eine Bereitstellung fehlschlägt, wenn ein Lebenszyklusereignis nicht gestartet wird	Jede unterstützte Region: 5	Nein	Maximale Anzahl von Minuten, bis eine Bereitstellung fehlschlägt, wenn ein Lebenszyklusereignis nicht beginnt, nachdem (1) eine Bereitstellung mithilfe der Konsole oder des AWS CLI-Befehls create-deployment ausgelöst wurde oder (2) das vorherige Lebenszyklusereignis abgeschlossen ist.
Anzahl der Bereitstellungsgruppen, die einem Amazon-ECS-Service zugeordnet werden können	Jede unterstützte Region: 1	Nein	Maximale Anzahl der Bereitstellungsgruppen, die einem Amazon-ECS-Service zugeordnet werden können
Anzahl der Instanzen, die an die API-Aktion übergeben werden können BatchGetOnPremisesInstances	Jede unterstützte Region: 100	Nein	Maximale Anzahl von Instanzen, die an die BatchGetOnPremisesInstances API-Aktion übergeben werden können

Name	Standard	Anpas	Description
Anzahl der Instances pro Konto, die von gleichzeitig ausgeführten Bereitstellungen verwendet werden	us-east-1: 3 000 Jede der anderen unterstützten Regionen: 1 000	Ja	Maximale Anzahl von Instances, die von gleichzeitigen Bereitstellungen verwendet werden können, die ausgeführt werden und mit einem Konto verknüpft sind
Anzahl der Listener für eine Datenverkehrsroute während einer Amazon-ECS-Bereitstellung	Jede unterstützte Region: 1	Nein	Die maximale Anzahl der Listener für eine Datenverkehrsroute während einer Amazon-ECS-Bereitstellung
Sekunden, bis ein Bereitstellungslebenszyklusereignis fehlschlägt, wenn es nicht abgeschlossen wird	Jede unterstützte Region: 3 600 Sekunden	Nein	Maximale Anzahl der Sekunden, bis ein Bereitstellungslebenszyklusereignis fehlschlägt, wenn es nicht abgeschlossen wird
Größe des Bereitstellungsgruppennamens	Jede unterstützte Region: 100	Nein	Maximale Anzahl von Zeichen im Namen einer Bereitstellungsgruppe
Größe des Tag-Schlüssels	Jede unterstützte Region: 128	Nein	Maximale Anzahl von Zeichen in einem Tag-Schlüssel
Größe des Tag-Werts	Jede unterstützte Region: 256	Nein	Maximale Anzahl von Zeichen in einem Tag-Wert

Name	Standard	Anpas	Description
Tags in einer Bereitstellungsgruppe	Jede unterstützte Region: 10	Nein	Maximale Anzahl von Tags in einer Bereitstellungsgruppe
Datenverkehr, der während einer AWS Lambda-Bereitstellung in einem Schritt verschoben werden kann	Jede unterstützte Region: 99	Nein	Maximaler Prozentsatz des Datenverkehrs, der während einer AWS Lambda-Bereitstellung schrittweise verschoben werden kann

Problembehebung CodeDeploy

Verwenden Sie die Themen in diesem Abschnitt, um Probleme und Fehler zu lösen, die bei der Verwendung auftreten können CodeDeploy.

> **Note**
>
> Sie können die Ursachen für viele Bereitstellungsfehler identifizieren, indem Sie die während der Bereitstellung erstellten Protokolldateien überprüfen. Der Einfachheit halber empfehlen wir, Amazon CloudWatch Logs zu verwenden, um Protokolldateien zentral zu überwachen, anstatt sie instanzweise anzuzeigen. Weitere Informationen finden Sie unter Monitoring Deployments with Amazon CloudWatch Tools.

Themen

- Allgemeine Probleme bei der Fehlerbehebung
- Beheben Sie Probleme bei der EC2/On-Premises Bereitstellung
- Beheben von Amazon ECS-Bereitstellungsproblemen
- Fehlerbehebung AWS Probleme bei der Lambda-Bereitstellung
- Beheben von Problemen bei Bereitstellungsgruppen
- Beheben von Instance-Problemen
- Beheben Sie GitHub Token-Probleme
- Probleme mit Amazon EC2 Auto Scaling beheben
- Fehlercodes für AWS CodeDeploy

Allgemeine Probleme bei der Fehlerbehebung

Themen

- Allgemeine Checkliste für die Fehlerbehebung
- CodeDeploy Bereitstellungsressourcen werden nur in einigen Fällen unterstützt AWS Regionen
- Die Verfahren in diesem Handbuch stimmen nicht mit der CodeDeploy Konsole überein
- Erforderliche IAM-Rollen sind nicht verfügbar

- Die Verwendung einiger Texteditoren zum Erstellen von AppSpec Dateien und Shell-Skripts kann dazu führen, dass Bereitstellungen fehlschlagen
- Verwenden von Finder in macOS zur Bündelung einer Anwendungsrevision kann zum Fehlschlagen von Bereitstellungen führen

Allgemeine Checkliste für die Fehlerbehebung

Sie können die folgende Checkliste zur Fehlerbehebung bei einer fehlgeschlagenen Bereitstellung verwenden.

1. Weitere Informationen, um den Grund für die fehlgeschlagene Bereitstellung zu ermitteln, finden Sie unter CodeDeploy Bereitstellungsdetails anzeigen und View Instance Details. Wenn Sie die Ursache nicht ermitteln können, überprüfen Sie die Elemente in dieser Checkliste.
2. Überprüfen Sie, ob Sie die Instances korrekt konfiguriert haben:
   - Wurde die Instance mit einem angegebenen EC2-Schlüsselpaar gestartet? Weitere Informationen finden Sie unter EC2-Schlüsselpaare im Amazon EC2 EC2-Benutzerhandbuch.
   - Ist das richtige IAM-Instance-Profil an die Instance angehängt? Weitere Informationen erhalten Sie unter Konfigurieren Sie eine Amazon EC2 EC2-Instance, mit der Sie arbeiten können CodeDeploy und Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances.
   - Wurde die Instance getaggt? Weitere Informationen finden Sie unter Arbeiten mit Tags in der Konsole im Amazon EC2 EC2-Benutzerhandbuch.
   - Ist der CodeDeploy Agent auf der Instance installiert, aktualisiert und läuft er? Weitere Informationen finden Sie unter Verwaltung des CodeDeploy Agentenbetriebs. Informationen darüber, welche Version des Agenten installiert ist, finden Sie unterErmitteln Sie die Version des Agenten CodeDeploy.
3. Prüfen Sie die Einstellungen für die Anwendung und für Bereitstellungsgruppen:
   - Informationen zum Prüfen der Anwendungseinstellungen finden Sie unter Anwendungsdetails anzeigen mit CodeDeploy.
   - Informationen zum Prüfen der Bereitstellungsgruppeneinstellungen finden Sie unter Details zur Bereitstellungsgruppe anzeigen mit CodeDeploy.
4. Überprüfen Sie, ob die Anwendungsrevision ordnungsgemäß konfiguriert ist:

- Überprüfen Sie das Format Ihrer AppSpec Datei. Weitere Informationen erhalten Sie unter Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy und CodeDeploy AppSpec Dateiverweis.
- Überprüfen Sie Ihren Amazon S3 S3-Bucket oder Ihr Amazon GitHub S3-Repository, um sicherzustellen, dass sich Ihre Anwendungsversion am erwarteten Speicherort befindet.
- Überprüfen Sie die Details Ihrer CodeDeploy Anwendungsrevision, um sicherzustellen, dass sie korrekt registriert ist. Weitere Informationen finden Sie unter Revisionsdetails der Anwendung anzeigen mit CodeDeploy.
- Wenn Sie von Amazon S3 aus bereitstellen, überprüfen Sie Ihren Amazon S3 S3-Bucket, um sicherzustellen, CodeDeploy dass Ihnen Berechtigungen zum Herunterladen der Anwendungsversion erteilt wurden. Weitere Informationen zu Bucket-Richtlinien finden Sie unter Voraussetzungen für die Bereitstellung.
- Wenn Sie von dort aus bereitstellen GitHub, überprüfen Sie in Ihrem GitHub Repository, ob CodeDeploy Ihnen Berechtigungen zum Herunterladen der Anwendungsversion erteilt wurden. Weitere Informationen erhalten Sie unter Erstellen Sie eine Bereitstellung mit CodeDeploy und GitHub Authentifizierung mit Anwendungen in CodeDeploy.

5. Überprüfen Sie, ob die Servicerolle ordnungsgemäß konfiguriert ist. Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.
6. Vergewissern Sie sich, dass Sie die folgenden Schritte in Erste Schritte mit CodeDeploy ausgeführt haben:
   - Hat einem Benutzer die entsprechenden Berechtigungen zugewiesen.
   - Installieren oder aktualisieren und konfigurieren Sie die AWS CLI.
   - Erstellen Sie ein IAM-Instanzprofil und eine Servicerolle.

   Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für AWS CodeDeploy.
7. Vergewissern Sie sich, dass Sie AWS CLI Version 1.6.1 oder höher verwenden. Zur Überprüfung der installierten Version rufen Sie aws --version auf.

Wenn Sie die fehlgeschlagene Bereitstellung noch immer nicht beheben können, prüfen Sie die anderen Probleme in diesem Thema.

CodeDeploy Bereitstellungsressourcen werden nur in einigen Fällen unterstützt AWS Regionen

Wenn Sie in der oder der CodeDeploy Konsole keine Anwendungen, Bereitstellungsgruppen, Instanzen oder andere Bereitstellungsressourcen sehen AWS CLI oder nicht darauf zugreifen können, stellen Sie sicher, dass Sie auf eine der unter Region und Endpunkte in aufgeführten AWS Regionen verweisen. Allgemeine AWS-Referenz

EC2-Instances und Amazon EC2 Auto Scaling Scaling-Gruppen, die in CodeDeploy Bereitstellungen verwendet werden, müssen in einer dieser Regionen gestartet und erstellt werden. AWS

Wenn Sie den verwenden AWS CLI, führen Sie den aws configure Befehl über den aus. AWS CLI Dann können Sie Ihre AWS Standardregion anzeigen und festlegen.

Wenn Sie die CodeDeploy Konsole verwenden, wählen Sie in der Navigationsleiste in der Regionsauswahl eine der unterstützten AWS Regionen aus.

> **Important**
>
> Um Dienste in der Region China (Peking) oder China (Ningxia) nutzen zu können, benötigen Sie ein Konto und Anmeldeinformationen für diese Regionen. Konten und Anmeldeinformationen für andere AWS Regionen funktionieren nicht für die Regionen Peking und Ningxia und umgekehrt.
> Informationen zu einigen Ressourcen für die Regionen China, wie z. B. CodeDeploy Resource Kit-Bucket-Namen und Verfahren zur CodeDeploy Agenteninstallation, sind in dieser Ausgabe des CodeDeploy Benutzerhandbuchs nicht enthalten.
> Weitere Informationen:
>
> - CodeDeployin Erste Schritte AWS in der Region China (Peking)
> - CodeDeploy Benutzerhandbuch für die Regionen China (englische Version | chinesische Version)

Die Verfahren in diesem Handbuch stimmen nicht mit der CodeDeploy Konsole überein

Die Verfahren in diesem Handbuch wurden geschrieben, um das neue Konsolendesign zu veranschaulichen. Wenn Sie die ältere Version der Konsole verwenden, gelten viele der Konzepte

und grundlegenden Prozeduren in diesem Handbuch weiterhin. Um auf die Hilfe in der neuen Konsole zuzugreifen, wählen Sie das Symbol für Informationen aus.

Erforderliche IAM-Rollen sind nicht verfügbar

Wenn Sie sich auf ein IAM-Instanzprofil oder eine Servicerolle verlassen, die als Teil eines AWS CloudFormation Stacks erstellt wurde, werden beim Löschen des Stacks auch alle IAM-Rollen gelöscht. Dies kann der Grund sein, warum die IAM-Rolle nicht mehr in der IAM-Konsole angezeigt wird und nicht CodeDeploy mehr wie erwartet funktioniert. Um dieses Problem zu beheben, müssen Sie die gelöschte IAM-Rolle manuell neu erstellen.

Die Verwendung einiger Texteditoren zum Erstellen von AppSpec Dateien und Shell-Skripts kann dazu führen, dass Bereitstellungen fehlschlagen

Einige Texteditoren fügen nicht konforme, nicht druckende Zeichen in Dateien ein. Wenn Sie Texteditoren verwenden, um AppSpec Dateien oder Shell-Skriptdateien für die Ausführung auf Amazon Linux-, Ubuntu Server- oder RHEL-Instances zu erstellen oder zu ändern, schlagen alle Bereitstellungen, die auf diesen Dateien basieren, möglicherweise fehl. Wenn Sie CodeDeploy diese Dateien während einer Bereitstellung verwenden, kann das Vorhandensein dieser Zeichen zu schwer zu behebenden Fehlern bei der AppSpec Dateiüberprüfung und der Skriptausführung führen.

Wählen Sie in der CodeDeploy Konsole auf der Seite mit den Ereignisdetails für die Bereitstellung die Option Protokolle anzeigen aus. (Oder Sie verwenden den Befehl get-deployment-instance AWS CLI , um den Befehl get-deployment-instance aufzurufen.) Suchen Sie nach Fehlern wie `invalid character`, `command not found` oder `file not found`.

Um dieses Problem zu beheben, empfehlen wir die folgenden Schritte:

- Verwenden Sie keine Texteditoren, die nicht druckbare Zeichen wie Zeilenumbrüche (`^M`Zeichen) in Ihre AppSpec Dateien und Shell-Skriptdateien einfügen.
- Verwenden Sie Texteditoren, die nicht druckbare Zeichen wie Zeilenumbrüche in Ihren AppSpec Dateien und Shell-Skriptdateien anzeigen, damit Sie alle eventuell eingeführten Zeichen finden und entfernen können. Beispiele für solche Texteditoren finden Sie, wenn Sie im Internet nach „Texteditor, der Wagenrücklauf anzeigen“ suchen.
- Verwenden Sie Texteditoren, die auf Amazon Linux-, Ubuntu Server- oder RHEL-Instances ausgeführt werden, um Shell-Skriptdateien zu erstellen, die auf Amazon Linux-, Ubuntu Server- oder RHEL-Instances ausgeführt werden. Beispiele für diese Art von Texteditoren finden Sie, wenn Sie im Internet nach „Linux Shell-Skript-Editor“ suchen.

- Wenn Sie einen Texteditor in Windows oder macOS verwenden müssen, um Shell-Skriptdateien für die Ausführung auf Amazon Linux-, Ubuntu Server- oder RHEL-Instances zu erstellen, verwenden Sie ein Programm oder Hilfsprogramm, das Text im Windows- oder macOS-Format in das Unix-Format konvertiert. Beispiele dieser Programme und Hilfsprogramme finden Sie, wenn Sie im Internet nach „DOS in UNIX“ oder „Mac zu UNIX“ suchen. Testen Sie die konvertierten Shell-Skriptdateien in den Zielbetriebssystemen.

Verwenden von Finder in macOS zur Bündelung einer Anwendungsrevision kann zum Fehlschlagen von Bereitstellungen führen

Bereitstellungen können fehlschlagen, wenn Sie die Finder-Anwendung mit grafischer Benutzeroberfläche (GUI) auf einem Mac verwenden, um eine AppSpec Datei und zugehörige Dateien und Skripts in einer Anwendungsrevisionsarchivdatei (.zip) zu bündeln (zip). Der Grund hierfür ist, dass Finder einen Zwischenordner `__MACOSX` in der ZIP-Datei erstellt und Komponentendateien darin platziert. CodeDeploy kann die Komponentendateien nicht finden, sodass die Bereitstellung fehlschlägt.

Um dieses Problem zu beheben, empfehlen wir Ihnen, den Befehl AWS CLI to call zu verwenden, der die Komponentendateien in die erwartete Struktur komprimiert. Alternativ können Sie die Komponentendateien mit Terminal anstelle der GUI zippen. Terminal erstellt keinen Zwischenordner `__MACOSX`.

Beheben Sie Probleme bei der EC2/On-Premises Bereitstellung

Themen

- CodeDeploy Fehler beim Fehlen der Anmeldeinformationen für das Plugin CommandPoller
- Die Bereitstellung schlägt fehl und es wird die Meldung „Die Validierung der mit PKCS7 signierten Nachricht ist fehlgeschlagen“ angezeigt
- Bereitstellung oder erneute Bereitstellung derselben Dateien in denselben Instance-Standorten schlägt fehl mit dem Fehler „The deployment failed because a specified file already exists at this location“
- Lange Dateipfade führen zu der Fehlermeldung „Keine solche Datei oder kein solches Verzeichnis“
- Long-running Prozesse können dazu führen, dass Bereitstellungen fehlschlagen
- Behebung eines fehlgeschlagenen AllowTraffic Lebenszyklusereignisses, bei dem kein Fehler in den Bereitstellungsprotokollen gemeldet wurde

- Behebung eines Fehlers oder eines ApplicationStop BeforeBlockTraffic Ereignisses im Lebenszyklus einer AfterBlockTraffic Bereitstellung
- Behebung eines fehlgeschlagenen DownloadBundle Deployment-Lifecycle-Ereignisses mit UnknownError: nicht zum Lesen geöffnet
- Durch Fehlerbehebung bei allen Lebenszyklusereignissen wurden Fehler übersprungen
- PowerShell Windows-Skripts verwenden standardmäßig nicht die 64-Bit-Version von Windows PowerShell

> **Note**
>
> Die Ursachen für viele Bereitstellungsfehler können identifiziert werden, indem Sie die während der Bereitstellung erstellten Protokolldateien überprüfen. Der Einfachheit halber empfehlen wir, Amazon CloudWatch Logs zu verwenden, um Protokolldateien zentral zu überwachen, anstatt sie instanzweise anzuzeigen. Weitere Informationen finden Sie unter CodeDeploy Logs in der CloudWatch Logs-Konsole anzeigen.

> **Tip**
>
> Ein Runbook, das viele Aufgaben zur Problembehandlung im Zusammenhang mit EC2/On-Premises Bereitstellungen automatisiert, finden Sie AWSSupport-TroubleshootCodeDeployin der Runbook-Referenz zu AWS Systems Manager Automation.

CodeDeploy Fehler beim Fehlen der Anmeldeinformationen für das Plugin CommandPoller

Wenn Sie eine Fehlermeldung wie `InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller: Missing credentials - please check if this instance was started with an IAM instance profile` erhalten, kann dies eine der folgenden Ursachen haben:

- Der Instanz, für die Sie die Bereitstellung durchführen, ist kein IAM-Instanzprofil zugeordnet.
- Für Ihr IAM-Instanzprofil sind nicht die richtigen Berechtigungen konfiguriert.

Ein IAM-Instance-Profil gewährt dem CodeDeploy Agenten die Erlaubnis, mit Amazon S3 zu kommunizieren CodeDeploy und Ihre Version von Amazon S3 herunterzuladen. Informationen zu EC2-Instances finden Sie unter Identitäts- und Zugriffsmanagement für AWS CodeDeploy. Informationen zu lokalen Instances finden Sie unter Working with On-Premises Instances.

Die Bereitstellung schlägt fehl und es wird die Meldung „Die Validierung der mit PKCS7 signierten Nachricht ist fehlgeschlagen“ angezeigt

Diese Fehlermeldung weist darauf hin, dass auf der Instanz eine Version des CodeDeploy Agenten ausgeführt wird, die nur den SHA-1 Hash-Algorithmus unterstützt. Die Support für den SHA-2 Hash-Algorithmus wurde in Version 1.0.1.854 des CodeDeploy Agenten eingeführt, die im November 2015 veröffentlicht wurde. Ab dem 17. Oktober 2016 schlagen Bereitstellungen fehl, wenn eine Version des CodeDeploy Agenten vor 1.0.1.854 installiert ist. Weitere Informationen finden Sie unter AWS So wechseln Sie zum SHA256-Hash-Algorithmus für SSL-Zertifikate, HINWEIS: CodeDeploy Server-Agents, die älter als Version 1.0.1.85 sind, außer Dienst stellen, und. Aktualisieren Sie den CodeDeploy Agenten

Bereitstellung oder erneute Bereitstellung derselben Dateien in denselben Instance-Standorten schlägt fehl mit dem Fehler „The deployment failed because a specified file already exists at this location“

Wenn CodeDeploy versucht wird, eine Datei für eine Instanz bereitzustellen, aber eine Datei mit demselben Namen bereits am angegebenen Zielspeicherort existiert, schlägt die Bereitstellung für diese Instanz möglicherweise fehl. Möglicherweise erhalten Sie die Fehlermeldung „Die Bereitstellung ist fehlgeschlagen, weil eine angegebene Datei bereits an diesem Speicherort vorhanden ist:*`location-name`*.“ Der Grund hierfür ist, dass CodeDeploy bei jeder Bereitstellung zuerst alle Dateien aus der vorherigen Bereitstellung löscht, die in einer Cleanup-Protokolldatei aufgelistet sind. Wenn sich in den Zielinstallationsordnern Dateien befinden, die nicht in dieser Bereinigungsdatei aufgeführt sind, interpretiert der CodeDeploy Agent dies standardmäßig als Fehler und schlägt bei der Bereitstellung fehl.

> **Note**
>
> Auf Amazon Linux-, RHEL- und Ubuntu Server-Instances befindet sich die Cleanup-Datei unter. `/opt/codedeploy-agent/deployment-root/deployment-instructions/`

Auf Windows Server-Instances lautet der Speicherort. `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\`

Am einfachsten lässt sich dieser Fehler verhindern, indem Sie eine andere Option als das Standardverhalten für das Fehlschlagen der Bereitstellung angeben. Für jede Bereitstellung können Sie wählen, ob die Bereitstellung fehlschlagen soll, die nicht in der Cleanup-Datei aufgeführten Dateien überschrieben werden oder die bereits in der Instance vorhandenen Dateien beibehalten werden sollen.

Die Option zum Überschreiben ist beispielsweise nützlich, wenn Sie nach der letzten Bereitstellung manuell eine Datei in einer Instance platziert haben, aber dann eine Datei mit demselben Namen zu der nächsten Anwendungsrevision hinzugefügt haben.

Sie können die Option zum Beibehalten für Dateien wählen, die Sie in der Instance platzieren, die Teil der nächsten Bereitstellung sein soll, ohne sie dem Anwendungsrevisionspaket hinzuzufügen. Die Option Retain ist auch nützlich, wenn sich Ihre Anwendungsdateien bereits in Ihrer Produktionsumgebung befinden und Sie sie CodeDeploy zum ersten Mal bereitstellen möchten. Weitere Informationen erhalten Sie unter Erstellen Sie eine EC2/On-Premises Compute Platform-Bereitstellung (Konsole) und Rollback-Verhalten bei vorhandenem Inhalt.

Problembehandlung **`Die Bereitstellung ist fehlgeschlagen, weil eine angegebene Datei bereits an diesem Speicherort vorhanden ist`**. Bereitstellungsfehler

Wenn Sie keine Option zum Überschreiben oder Beibehalten von Inhalten angeben, die CodeDeploy in Ihren Ziel-Bereitstellungsstandorten erkennt (oder wenn Sie keine Bereitstellungsoption für die Behandlung vorhandener Inhalte in einem programmatischen Befehl angeben), können Sie den Fehler beheben.

Die folgenden Informationen gelten nur, wenn Sie angeben, dass Inhalte nicht beibehalten oder überschrieben werden sollen.

Wenn Sie versuchen, Dateien mit denselben Namen und Speicherorten erneut bereitzustellen, ist es wahrscheinlicher, dass die erneute Bereitstellung erfolgreich ist, wenn Sie den Anwendungsnamen und die Bereitstellungsgruppe mit derselben zugrunde liegenden Bereitstellungsgruppen-ID angeben, die Sie zuvor verwendet haben. CodeDeploy verwendet die zugrunde liegende Bereitstellungsgruppen-ID, um Dateien zu identifizieren, die vor einer erneuten Bereitstellung entfernt werden müssen.

Das Bereitstellen neuer Dateien oder die erneute Bereitstellung der gleichen Dateien an denselben Standorten in Instances kann aus folgenden Gründen fehlschlagen:

- Sie haben einen anderen Anwendungsnamen für eine erneute Bereitstellung der gleichen Revision in denselben Instances angegeben. Die erneute Bereitstellung schlägt fehl, weil auch beim gleichen Bereitstellungsgruppennamen die Verwendung eines anderen Anwendungsnamens bedeutet, dass eine andere zugrunde liegende Bereitstellungsgruppen-ID verwendet wird.
- Sie haben eine Bereitstellungsgruppe für eine Anwendung gelöscht und neu erstellt und dann versucht, die gleiche Revision für die Bereitstellungsgruppe erneut bereitzustellen. Die erneute Bereitstellung schlägt fehl, da selbst wenn der Name der Bereitstellungsgruppe identisch ist, CodeDeploy auf eine andere zugrunde liegende Bereitstellungsgruppen-ID verwiesen wird.
- Sie haben eine Anwendung und eine Bereitstellungsgruppe in CodeDeploy gelöscht und dann eine neue Anwendung und Bereitstellungsgruppe mit denselben Namen wie die gelöschten erstellt. Danach haben Sie versucht, eine Revision, die für die vorherige Bereitstellungsgruppe bereitgestellt wurde, erneut für die neue mit demselben Namen bereitzustellen. Die erneute Bereitstellung schlägt fehl, da die Namen der Anwendung und der Bereitstellungsgruppe zwar identisch sind, aber CodeDeploy dennoch auf die ID der Bereitstellungsgruppe verwiesen wird, die Sie gelöscht haben.
- Sie haben eine Revision für eine Bereitstellungsgruppe bereitgestellt und dann dieselbe Revision für eine andere Bereitstellungsgruppe für dieselben Instances bereitgestellt. Die zweite Bereitstellung schlägt fehl, da CodeDeploy auf eine andere zugrunde liegende Bereitstellungsgruppen-ID verweist.
- Sie haben eine Revision für eine Bereitstellungsgruppe bereitgestellt und dann eine andere Revision für eine andere Bereitstellungsgruppe für dieselben Instances bereitgestellt. Es ist mindestens eine Datei mit demselben Namen an demselben Standort vorhanden, die in der zweiten Bereitstellungsgruppe bereitgestellt werden soll. Die zweite Bereitstellung schlägt fehl, weil die vorhandene Datei CodeDeploy nicht entfernt wird, bevor die zweite Bereitstellung gestartet wird. Beide Bereitstellungen verweisen auf unterschiedliche Bereitstellungsgruppen-IDs.
- Sie haben eine Revision in bereitgestellt CodeDeploy, aber es gibt mindestens eine Datei mit demselben Namen und am selben Speicherort. Die Bereitstellung schlägt fehl, da die vorhandene Datei standardmäßig CodeDeploy nicht entfernt wird, bevor die Bereitstellung gestartet wird.

Um diese Situationen zu beheben, führen Sie einen der folgenden Schritte aus:

- Entfernen Sie die Dateien aus den Standorten und Instances, für die sie zuvor bereitgestellt wurden, und versuchen Sie dann die Bereitstellung erneut.

- Geben Sie in der AppSpec Datei Ihrer Revision entweder bei den Ereignissen im Lebenszyklus ApplicationStop oder bei der BeforeInstall Bereitstellung ein benutzerdefiniertes Skript an, um Dateien an beliebigen Speicherorten zu löschen, die mit den Dateien übereinstimmen, die Ihre Revision gerade installieren wird.
- Stellen Sie die Dateien in Standorten oder Instances (erneut) bereit, die nicht Teil vorheriger Bereitstellungen waren.
- Bevor Sie eine Anwendung oder eine Bereitstellungsgruppe löschen, stellen Sie eine Revision bereit, die eine AppSpec Datei enthält, in der angegeben ist, dass keine Dateien in die Instanzen kopiert werden sollen. Geben Sie für die Bereitstellung den Anwendungsnamen und den Bereitstellungsgruppennamen an, denen dieselben Anwendungs- und Bereitstellungsgruppen-IDs zugrunde liegen wie denen, die Sie löschen möchten. (Sie können den Befehl get-deployment-group verwenden, um die Bereitstellungsgruppen-ID abzurufen.) CodeDeployverwendet die zugrunde liegende Bereitstellungsgruppen-ID und AppSpec -Datei, um alle Dateien zu entfernen, die bei der vorherigen erfolgreichen Bereitstellung installiert wurden.

Lange Dateipfade führen zu der Fehlermeldung „Keine solche Datei oder kein solches Verzeichnis“

Wenn Sie bei Bereitstellungen für Windows-Instanzen einen Dateipfad mit mehr als 260 Zeichen im Dateibereich Ihrer appspec.yml-Datei angeben, schlagen Bereitstellungen möglicherweise mit einer Fehlermeldung ähnlich der folgenden fehl:

`No such file or directory @ dir_s_mkdir - C:\`*`your-long-file-path`*

Dieser Fehler tritt auf, weil Windows standardmäßig keine Dateipfade mit mehr als 260 Zeichen zulässt, wie in der Dokumentation von Microsoft beschrieben.

Für CodeDeploy Agentenversionen 1.4.0 oder höher können Sie lange Dateipfade je nach Agenteninstallationsprozess auf zwei Arten aktivieren:

Wenn der CodeDeploy Agent noch nicht installiert wurde:

1. Aktivieren Sie auf dem Computer, auf dem Sie den CodeDeploy Agenten installieren möchten, den `LongPathsEnabled` Windows-Registrierungsschlüssel mit diesem Befehl:

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"
          -Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Installieren Sie den CodeDeploy Agenten. Weitere Informationen finden Sie unter Installieren Sie den CodeDeploy Agenten.

Wenn der CodeDeploy Agent bereits installiert wurde:

1. Aktivieren Sie auf dem CodeDeploy Agent-Computer den `LongPathsEnabled` Windows-Registrierungsschlüssel mit diesem Befehl:

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"
-Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Starten Sie den CodeDeploy Agenten neu, damit die Änderung des Registrierungsschlüssels wirksam wird. Verwenden Sie diesen Befehl, um den Agenten neu zu starten:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Long-running Prozesse können dazu führen, dass Bereitstellungen fehlschlagen

Wenn Sie bei Bereitstellungen auf Amazon Linux-, Ubuntu Server- und RHEL-Instances über ein Bereitstellungsskript verfügen, das einen lang andauernden Prozess startet, warten Sie CodeDeploy möglicherweise lange auf das Ereignis im Bereitstellungslebenszyklus und schlagen dann bei der Bereitstellung fehl. Der Grund hierfür ist: Wenn der Prozess länger läuft als die erwartete Dauer der Vorder- und Hintergrundprozesse in dem Ereignis, stoppt CodeDeploy die Bereitstellung und lässt sie fehlschlagen, auch wenn der Prozess noch wie erwartet ausgeführt wird.

Beispiel: Eine Anwendungsrevision enthält zwei Dateien im Stammverzeichnis, `after-install.sh` und `sleep.sh`. Die AppSpec Datei enthält die folgenden Anweisungen:

```
version: 0.0
os: linux
files:
  - source: ./sleep.sh
    destination: /tmp
hooks:
  AfterInstall:
    - location: after-install.sh
      timeout: 60
```

Die `after-install.sh` Datei wird während des AfterInstall Anwendungslebenszyklus ausgeführt. Der Inhalt lautet folgendermaßen:

```
#!/bin/bash
/tmp/sleep.sh
```

Die Datei `sleep.sh` enthält folgenden Code, der die Ausführung des Programms für drei Minuten (180 Sekunden) unterbricht. Dadurch werden einige lange laufende Prozesse simuliert:

```
#!/bin/bash
sleep 180
```

Bei `after-install.sh` Aufrufen `sleep.sh` `sleep.sh` startet und läuft drei Minuten (180 Sekunden), also zwei Minuten (120 Sekunden) nach CodeDeploy der `sleep.sh` erwarteten (und damit verbundenen`after-install.sh`) Unterbrechung der Ausführung. Nach Ablauf des Timeouts von einer Minute (60 Sekunden) wird die Bereitstellung zum Zeitpunkt des AfterInstall Anwendungslebenszyklus CodeDeploy angehalten und schlägt fehl, obwohl sie `sleep.sh` weiterhin wie erwartet ausgeführt wird. Der folgende Fehler wird angezeigt:

`Script at specified location: after-install.sh failed to complete in 60 seconds`.

Sie können nicht einfach ein kaufmännisches Und-Zeichen (&) in `after-install.sh` einfügen, um `sleep.sh` im Hintergrund auszuführen.

```
#!/bin/bash
# Do not do this.
/tmp/sleep.sh &
```

Dadurch kann die Bereitstellung bis zum standardmäßigen Timeout von einer Stunde im Status „Ausstehend“ verbleiben. Danach wird die Bereitstellung beim AfterInstall Anwendungslebenszyklus-Ereignis wie zuvor CodeDeploy beendet und schlägt fehl.

Rufen Sie in `after-install.sh` `sleep.sh` wie folgt auf, sodass CodeDeploy Sie nach dem Start des Prozesses weitermachen können:

```
#!/bin/bash
/tmp/sleep.sh > /dev/null 2> /dev/null < /dev/null &
```

Im vorherigen Aufruf ist `sleep.sh` der Name des Prozesses, der im Hintergrund ausgeführt werden soll, wobei stdout, stderr und stdin nach `/dev/null` umgeleitet werden.

Behebung eines fehlgeschlagenen AllowTraffic Lebenszyklusereignisses, bei dem kein Fehler in den Bereitstellungsprotokollen gemeldet wurde

In einigen Fällen schlägt eine blue/green Bereitstellung während des AllowTraffic Lebenszyklusereignisses fehl, aber die Bereitstellungsprotokolle geben keinen Hinweis auf die Ursache des Fehlers.

Dieser Fehler ist in der Regel auf falsch konfigurierte Zustandsprüfungen in Elastic Load Balancing für den Classic Load Balancer, Application Load Balancer oder Network Load Balancer zurückzuführen, der zur Verwaltung des Datenverkehrs für die Bereitstellungsgruppe verwendet wird.

Um das Problem zu lösen, überprüfen Sie die Konfiguration der Zustandsprüfung für den Load Balancer und korrigieren Sie eventuelle Fehler.

Informationen zu Classic Load Balancers finden Sie unter Configure Health Checks im Benutzerhandbuch für Classic Load Balancers und ConfigureHealthCheckin der Elastic Load Balancing API-Referenzversion 2012-06-01.

Informationen zu Application Load Balancers finden Sie unter Health Checks für Ihre Zielgruppen im Benutzerhandbuch für Application Load Balancers.

Informationen zu Network Load Balancers finden Sie unter Health Checks für Ihre Zielgruppen im Network Load Balancer User Guide.

Behebung eines Fehlers oder eines ApplicationStop BeforeBlockTraffic Ereignisses im Lebenszyklus einer AfterBlockTraffic Bereitstellung

Während einer Bereitstellung führt der CodeDeploy Agent die Skripts aus, die für ApplicationStop BeforeBlockTraffic, und AfterBlockTraffic in der AppSpec Datei aus der vorherigen erfolgreichen Bereitstellung angegeben sind. (Alle anderen Skripts werden von der AppSpec Datei in der aktuellen Bereitstellung aus ausgeführt.) Wenn eines dieser Skripts einen Fehler enthält und nicht erfolgreich ausgeführt wird, kann Bereitstellung fehlschlagen.

Mögliche Gründe für diese Ausfälle sind:

- Der CodeDeploy Agent findet die *deployment-group-id*_last_successful_install Datei am richtigen Speicherort, aber der in der *deployment-group-id*_last_successful_install Datei aufgeführte Speicherort ist nicht vorhanden.

  Auf Amazon Linux-, Ubuntu Server- und RHEL-Instances muss diese Datei in `/opt/codedeploy-agent/deployment-root/deployment-instructions` vorhanden sein.

  Auf Windows Server-Instances muss diese Datei im `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions` Ordner gespeichert werden.
- An dem in der *deployment-group-id*_last_successful_install Datei angegebenen Speicherort ist entweder die AppSpec Datei ungültig oder die Skripts werden nicht erfolgreich ausgeführt.
- Das Skript enthält einen Fehler, der nicht korrigiert werden kann, sodass es nie erfolgreich ausgeführt werden wird.

Verwenden Sie die CodeDeploy Konsole, um zu untersuchen, warum eine Bereitstellung bei einem dieser Ereignisse möglicherweise fehlgeschlagen ist. Wählen Sie auf der Detailseite für die Bereitstellung View events (Ereignisse anzeigen) aus. Wählen Sie auf der Detailseite für die Instance in der AfterBlockTrafficZeile ApplicationStopBeforeBlockTraffic, oder die Option Logs anzeigen aus. Oder verwenden Sie den AWS CLI , um den Befehl get-deployment-instance aufzurufen.

Wenn die Ursache des Fehlers ein Skript aus der letzten erfolgreichen Bereitstellung ist, das nie erfolgreich ausgeführt wird, erstellen Sie eine Bereitstellung und geben Sie an, dass die AfterBlockTraffic Fehler ApplicationStop BeforeBlockTraffic, und ignoriert werden sollen. Es gibt zwei Möglichkeiten dafür:

- Verwenden Sie die CodeDeploy Konsole, um eine Bereitstellung zu erstellen. Wählen Sie auf der Seite Bereitstellung erstellen unter ApplicationStop Lifecycle-Event Failure die Option Don't fail the deployment to an instance if this Lifecycle event on the instance failed if this Lifecycle Event in the Instance failed.
- Verwenden Sie den AWS CLI , um den create-deployment Befehl aufzurufen und die `--ignore-application-stop-failures` Option einzufügen.

Wenn Sie die Anwendungsrevision erneut bereitstellen, wird die Bereitstellung fortgesetzt, auch wenn eines dieser drei Lebenszyklusereignisse fehlschlägt. Wenn die neue Revision korrigierte Skripts für diese Lebenszyklusereignisse umfasst, können künftige Bereitstellungen erfolgreich ausgeführt werden, ohne diese Korrektur anzuwenden.

Behebung eines fehlgeschlagenen DownloadBundle Deployment-Lifecycle-Ereignisses mit UnknownError: nicht zum Lesen geöffnet

Wenn Sie versuchen, eine Anwendungsrevision von Amazon S3 aus bereitzustellen, und die Bereitstellung während des Bereitstellungslebenszyklus mit dem folgenden `UnknownError: not opened for reading` Fehler fehlschlägt: DownloadBundle

- Es gab einen internen Amazon S3 S3-Servicefehler. Stellen Sie die Anwendungsrevision erneut bereit.
- Das IAM-Instance-Profil auf Ihrer EC2-Instance hat keine Berechtigungen für den Zugriff auf die Anwendungsversion in Amazon S3. Informationen zu den Amazon S3 S3-Bucket-Richtlinien finden Sie unter Eine Revision CodeDeploy auf Amazon S3 übertragen (nur EC2/On-Premises Bereitstellungen) undVoraussetzungen für die Bereitstellung.
- Die Instances, für die Sie bereitstellen, sind einer AWS Region zugeordnet (z. B. USA West (Oregon)), aber der Amazon S3 S3-Bucket, der die Anwendungsrevision enthält, ist einer anderen AWS Region zugeordnet (z. B. USA Ost (Nord-Virginia)). Stellen Sie sicher, dass sich die Anwendungsversion in einem Amazon S3 S3-Bucket befindet, der derselben AWS Region wie die Instances zugeordnet ist.

Wählen Sie auf der Ereignisdetail-Seite für die Bereitstellung auf der Zeile Download bundle (Bundle herunterladen) die Option View logs (Protokolle anzeigen). Oder verwenden Sie den AWS CLI , um den Befehl get-deployment-instance aufzurufen. Wenn dieser Fehler aufgetreten ist, sollte die Ausgabe einen Fehler mit dem Fehlercode `UnknownError` und die Fehlermeldung `not opened for reading` enthalten.

So ermitteln Sie den Grund für diesen Fehler:

1. Aktivieren Sie die Übertragungsprotokollierung in mindestens einer der Instances und stellen Sie dann die Anwendungsrevision erneut bereit.
2. Überprüfen Sie die Übertragungsprotokollierungsdatei, um den Fehler zu ermitteln. Allgemeine Fehlermeldungen für dieses Problem enthalten den Ausdruck „access denied“.
3. Nachdem Sie die Protokolldateien geprüft haben, sollten Sie die Übertragungsprotokollierung deaktivieren, um die Größe der Protokolldatei und die Menge der vertraulichen Informationen zu reduzieren, die zukünftig unverschlüsselt in der Ausgabe auf der Instance erscheinen.

Informationen darüber, wie Sie die Wire-Logging-Datei finden und die Wire-Logging-Funktion aktivieren und deaktivieren, finden Sie `:log_aws_wire:` in der Referenz zur CodeDeploy Agentenkonfiguration.

Durch Fehlerbehebung bei allen Lebenszyklusereignissen wurden Fehler übersprungen

Wenn alle Lebenszyklusereignisse einer EC2- oder lokalen Bereitstellung übersprungen werden, erhalten Sie möglicherweise eine ähnliche Fehlermeldung wie. `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)` Hier finden Sie einige möglichen Ursachen und Lösungen:

- Der CodeDeploy Agent ist möglicherweise nicht auf der Instance installiert oder läuft nicht. So stellen Sie fest, ob der CodeDeploy Agent läuft:
  - Unter Amazon Linux RHEL oder Ubuntu Server führen Sie die folgenden Schritte aus:

    ```
    systemctl status codedeploy-agent
    ```

  - Unter Windows führen Sie die folgenden Schritte aus:

    ```
    powershell.exe -Command Get-Service -Name CodeDeployagent
    ```

  Wenn der CodeDeploy Agent nicht installiert ist oder nicht ausgeführt wird, finden Sie weitere Informationen unterStellen Sie sicher, dass der CodeDeploy Agent läuft.

  Ihre Instance ist möglicherweise nicht in der Lage, den öffentlichen Endpunkt CodeDeploy oder den öffentlichen Amazon S3 S3-Endpunkt über Port 443 zu erreichen. Führen Sie einen der folgenden Schritte aus:

  - Weisen Sie der Instance eine öffentliche IP-Adresse zu und verwenden Sie ihre Routing-Tabelle, um den Internetzugriff zu ermöglichen. Stellen Sie sicher, dass die der Instance zugeordnete Sicherheitsgruppe den ausgehenden Zugriff über Port 443 (HTTPS) zulässt. Weitere Informationen finden Sie unter Kommunikationsprotokoll und Port für den CodeDeploy Agenten.

  - Wenn eine Instance in einem privaten Subnetz bereitgestellt wird, verwenden Sie einen NAT-Gateway anstelle eines Internet-Gateways in der Routing-Tabelle. Weitere Informationen finden Sie unter NAT-Gateways.
- Für die Servicerolle für sind CodeDeploy möglicherweise keine Berechtigungen erforderlich. Informationen zum Konfigurieren einer Servicerolle für CodeDeploy finden Sie unter Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.
- Wenn Sie einen HTTP-Proxy verwenden, stellen Sie sicher, dass dieser in der `:proxy_uri:` Einstellung in der CodeDeploy Agentenkonfigurationsdatei angegeben ist. Weitere Informationen finden Sie unter CodeDeploy Referenz zur Agentenkonfiguration.
- Die Datums- und Uhrzeitsignatur Ihrer Bereitstellungs-Instance stimmt möglicherweise nicht mit der Signatur für Datum und Uhrzeit Ihrer Bereitstellungsanfrage überein. Suchen Sie nach einem ähnlichen Fehler wie `Cannot reach InstanceService: Aws::CodeDeployCommand::Errors::InvalidSignatureException - Signature expired` in Ihrer CodeDeploy Agenten-Protokolldatei. Wenn dieser Fehler angezeigt wird, befolgen Sie die Schritte in Problembehandlung bei Bereitstellungsfehlern „InvalidSignatureException — Signatur ist abgelaufen: [Zeit] ist jetzt vor [Zeit]“. Weitere Informationen finden Sie unter Protokolldaten für CodeDeploy EC2/On-Premises Bereitstellungen anzeigen.
- Der CodeDeploy Agent wird möglicherweise nicht mehr ausgeführt, weil einer Instanz nicht mehr genügend Arbeitsspeicher oder Festplattenspeicher zur Verfügung steht. Versuchen Sie, die Anzahl der archivierten Bereitstellungen auf Ihrer Instanz zu verringern, indem Sie die `max_revisions` Einstellung in der CodeDeploy Agentenkonfiguration aktualisieren. Wenn Sie dies für eine EC2-Instance tun und das Problem weiterhin besteht, sollten Sie die Verwendung einer größeren Instance in Betracht ziehen. Beispiel: Wenn Ihr Instance-Typ `t2.small` ist, verwenden Sie `t2.medium`. Weitere Informationen finden Sie unter Vom Agenten installierte Dateien CodeDeployCodeDeploy Referenz zur Agentenkonfiguration, und Instanztypen.
- Der Instance, für die Sie die Bereitstellung durchführen, ist möglicherweise kein IAM-Instanzprofil angehängt, oder es ist ein IAM-Instanzprofil angehängt, das nicht über die erforderlichen Berechtigungen verfügt.
  - Wenn Ihrer Instanz kein IAM-Instanzprofil angehängt ist, erstellen Sie eines mit den erforderlichen Berechtigungen und hängen Sie es dann an.
  - Wenn Ihrer Instance bereits ein IAM-Instanzprofil angehängt ist, stellen Sie sicher, dass es über die erforderlichen Berechtigungen verfügt.

Nachdem Sie überprüft haben, dass Ihr zugeordnetes Instance-Profil mit den erforderlichen Berechtigungen konfiguriert ist, starten Sie Ihre Instance neu. Weitere Informationen finden Sie unter Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances und IAM-Rollen für Amazon EC2 im Amazon EC2 EC2-Benutzerhandbuch.

PowerShell Windows-Skripts verwenden standardmäßig nicht die 64-Bit-Version von Windows PowerShell

Wenn ein PowerShell Windows-Skript, das als Teil einer Bereitstellung ausgeführt wird, auf 64-Bit-Funktionalität angewiesen ist (z. B. weil es mehr Speicher beansprucht, als eine 32-Bit-Anwendung zulässt, oder Bibliotheken aufruft, die nur in einer 64-Bit-Version angeboten werden), stürzt das Skript möglicherweise ab oder wird nicht wie erwartet ausgeführt. Das liegt daran, dass standardmäßig die 32-Bit-Version von Windows CodeDeploy verwendet wird, PowerShell um PowerShell Windows-Skripts auszuführen, die Teil einer Anwendungsrevision sind.

Fügen Sie am Anfang jedes Skripts, das mit der 64-Bit-Version von Windows ausgeführt werden muss, Code wie den folgenden hinzu 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...
# ...
```

Obwohl die Dateipfadinformationen in diesem Code nicht intuitiv erscheinen mögen, PowerShell verwendet 32-Bit-Windows einen Pfad wie den folgenden:

`c:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe`

64-Bit-Windows PowerShell verwendet einen Pfad wie:

`c:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`

Beheben von Amazon ECS-Bereitstellungsproblemen

Themen

- Beim Warten auf den Satz einer Ersatzaufgabe tritt ein Timeout auf
- Beim Warten auf die Fortsetzung einer Benachrichtigung tritt ein Timeout auf
- Die IAM-Rolle verfügt nicht über genügend Berechtigungen
- Beim Warten auf einen Statusrückruf ist bei der Bereitstellung ein Timeout aufgetreten
- Die Bereitstellung ist fehlgeschlagen, weil eine oder mehrere der Funktionen zur Überprüfung von Lebenszyklusereignissen fehlgeschlagen sind
- Der ELB konnte aufgrund des folgenden Fehlers nicht aktualisiert werden: Die primäre Taskset-Zielgruppe muss sich hinter dem Listener befinden
- Meine Bereitstellung schlägt manchmal fehl, wenn Auto Scaling verwendet wird
- Nur ALB unterstützt das schrittweise Routing des Datenverkehrs. Verwenden Sie stattdessen das AllAtOnce Traffic-Routing, wenn Sie create/update eine Gruppe einrichten
- Obwohl meine Bereitstellung erfolgreich war, besteht der Ersatz-Tasksatz bei den Elastic Load Balancing-Zustandsprüfungen nicht und meine Anwendung ist ausgefallen
- Kann ich einer Bereitstellungsgruppe mehrere Load Balancer zuordnen?
- Kann ich CodeDeploy blue/green Bereitstellungen ohne einen Load Balancer durchführen?
- Wie kann ich meinen Amazon ECS-Service während einer Bereitstellung mit neuen Informationen aktualisieren?

Beim Warten auf den Satz einer Ersatzaufgabe tritt ein Timeout auf

Problem: Sie sehen die folgende Fehlermeldung bei der Bereitstellung Ihrer Amazon ECS-Anwendung mit CodeDeploy:

```
The deployment timed out while waiting for the replacement task set to become healthy. This time out period is 60 minutes.
```

Mögliche Ursache: Dieser Fehler kann auftreten, wenn Ihre Aufgabendefinitionsdatei oder andere bereitstellungsbezogene Dateien einen Fehler enthalten. Wenn das `image` Feld in Ihrer Aufgabendefinitionsdatei beispielsweise einen Tippfehler enthält, versucht Amazon ECS, das falsche Container-Image abzurufen und schlägt ständig fehl, was diesen Fehler verursacht.

Mögliche Korrekturen und nächste Schritte:

- Korrigieren Sie Tippfehler und Konfigurationsprobleme in Ihrer Aufgabendefinitionsdatei und anderen Dateien.
- Sehen Sie sich das entsprechende Amazon ECS-Serviceereignis an und finden Sie heraus, warum Ersatzaufgaben nicht mehr ordnungsgemäß ausgeführt werden. Weitere Informationen zu Amazon ECS-Ereignissen finden Sie unter Amazon ECS-Ereignisse im Amazon Elastic Container Service Developer Guide.
- Im Amazon Elastic Container Service Developer Guide finden Sie im Abschnitt zur Amazon ECS-Fehlerbehebung Informationen zu Fehlern im Zusammenhang mit den Meldungen in der Veranstaltung.

Beim Warten auf die Fortsetzung einer Benachrichtigung tritt ein Timeout auf

Problem: Sie sehen die folgende Fehlermeldung bei der Bereitstellung Ihrer Amazon ECS-Anwendung mit CodeDeploy:

```
The deployment timed out while waiting for a notification to continue. This time out period is n minutes.
```

Mögliche Ursache: Dieser Fehler kann auftreten, wenn Sie bei der Erstellung Ihrer Bereitstellungsgruppe im Feld Geben Sie an, wann der Verkehr umgeleitet werden soll, eine Wartezeit angegeben haben, die Bereitstellung jedoch nicht abgeschlossen werden konnte, bevor die Wartezeit abgelaufen ist.

Mögliche Problembehebungen und nächste Schritte:

- Stellen Sie in Ihrer Bereitstellungsgruppe die Option Geben Sie an, wann der Datenverkehr umgeleitet werden soll, auf einen längeren Zeitraum ein und führen Sie die Bereitstellung erneut durch. Weitere Informationen finden Sie unter Eine Bereitstellungsgruppe für eine Amazon ECS-Bereitstellung erstellen (Konsole).
- Ändern Sie in Ihrer Bereitstellungsgruppe die Option Geben Sie an, wann der Verkehr umgeleitet werden soll in Verkehr sofort umleiten und erneut bereitstellen. Weitere Informationen finden Sie unter Eine Bereitstellungsgruppe für eine Amazon ECS-Bereitstellung erstellen (Konsole).
- Stellen Sie es erneut bereit und führen Sie dann den AWS CLI Befehl aws deploy continue-deployment mit der Option auf aus. `--deployment-wait-type READY_WAIT` Stellen Sie sicher, dass Sie diesen Befehl ausführen, bevor die unter Geben Sie an, wann der Verkehr umgeleitet werden soll, angegebene Zeit abläuft.

Die IAM-Rolle verfügt nicht über genügend Berechtigungen

Problem: Sie sehen die folgende Fehlermeldung bei der Bereitstellung Ihrer Amazon ECS-Anwendung mit: CodeDeploy

```
The IAM role role-arn does not give you permission to perform operations in the following AWS service: AWSLambda.
```

Mögliche Ursache: Dieser Fehler kann auftreten, wenn Sie im `Hooks` Abschnitt der AppSpec Datei eine Lambda-Funktion angegeben, dem Lambda-Service jedoch keine CodeDeploy Genehmigung erteilt haben.

Mögliche Lösung: Fügen Sie die `lambda:InvokeFunction` Berechtigung zur CodeDeploy Servicerolle hinzu. Um diese Berechtigung hinzuzufügen, fügen Sie der Rolle eine der AWS folgenden verwalteten Richtlinien hinzu: **`AWSCodeDeployRoleForECS`** oder**`AWSCodeDeployRoleForECSLimited`**. Informationen zu diesen Richtlinien und dazu, wie Sie sie der CodeDeploy Servicerolle hinzufügen, finden Sie unterSchritt 2: Erstellen Sie eine Servicerolle für CodeDeploy.

Beim Warten auf einen Statusrückruf ist bei der Bereitstellung ein Timeout aufgetreten

Problem: Sie sehen die folgende Fehlermeldung bei der Bereitstellung Ihrer Amazon ECS-Anwendung mit: 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.
```

Mögliche Ursache: Dieser Fehler kann auftreten, wenn Sie im `Hooks` Abschnitt der AppSpec Datei eine Lambda-Funktion angegeben haben, die Lambda-Funktion jedoch nicht die erforderliche `PutLifecycleEventHookExecutionStatus` API aufrufen konnte, an die ein `Succeeded` `Failed` OR-Status zurückgegeben werden sollte. CodeDeploy

Mögliche Korrekturen und nächste Schritte:

- Fügen Sie die `codedeploy:putlifecycleEventHookExecutionStatus` Berechtigung zur Lambda-Ausführungsrolle hinzu, die von der Lambda-Funktion verwendet wird, die Sie in der AppSpec Datei angegeben haben. Diese Berechtigung gewährt der Lambda-Funktion die Möglichkeit, den Status oder an zurückzugeben. `Succeeded` `Failed` CodeDeploy Weitere Informationen zur Lambda-Ausführungsrolle finden Sie unter Lambda-Ausführungsrolle im AWS Lambda Benutzerhandbuch.
- Überprüfen Sie Ihren Lambda-Funktionscode und die Ausführungsprotokolle, um sicherzustellen, dass Ihre Lambda-Funktion die `PutLifecycleEventHookExecutionStatus` API aufruft CodeDeploy, um zu CodeDeploy erfahren, ob der Lifecycle-Validierungstest oder `Succeeded` `Failed` Informationen zur `putlifecycleEventHookExecutionStatus` API finden Sie PutLifecycleEventHookExecutionStatus in der AWS CodeDeploy API-Referenz. Informationen zu Lambda-Ausführungsprotokollen finden Sie unter Zugreifen auf CloudWatch Amazon-Protokolle für AWS Lambda.

Die Bereitstellung ist fehlgeschlagen, weil eine oder mehrere der Funktionen zur Überprüfung von Lebenszyklusereignissen fehlgeschlagen sind

Problem: Sie sehen die folgende Fehlermeldung bei der Bereitstellung Ihrer Amazon ECS-Anwendung mit CodeDeploy:

```
The deployment failed because one or more of the lifecycle event validation
functions failed.
```

Mögliche Ursache: Dieser Fehler kann auftreten, wenn Sie im `Hooks` Abschnitt der AppSpec Datei eine Lambda-Funktion angegeben haben, die Lambda-Funktion jedoch CodeDeploy beim Aufruf

wieder `Failed` aktiviert wurde. `PutLifecycleEventHookExecutionStatus` Dieser Fehler weist darauf hin CodeDeploy , dass der Lifecycle-Validierungstest fehlgeschlagen ist.

Möglicher nächster Schritt: Überprüfen Sie Ihre Lambda-Ausführungsprotokolle, um festzustellen, warum der Validierungstestcode fehlschlägt. Informationen zu Lambda-Ausführungsprotokollen finden Sie unter Zugreifen auf CloudWatch Amazon-Protokolle für. AWS Lambda

Der ELB konnte aufgrund des folgenden Fehlers nicht aktualisiert werden: Die primäre Taskset-Zielgruppe muss sich hinter dem Listener befinden

Problem: Sie sehen die folgende Fehlermeldung bei der Bereitstellung Ihrer Amazon ECS-Anwendung mit: CodeDeploy

```
The ELB could not be updated due to the following error: Primary taskset target group must be behind listener
```

Mögliche Ursache: Dieser Fehler kann auftreten, wenn Sie einen optionalen Test-Listener konfiguriert haben und dieser mit der falschen Zielgruppe konfiguriert ist. Weitere Hinweise zum Test-Listener finden Sie unter CodeDeploy Bevor Sie mit einer Amazon-ECS-Bereitstellung beginnen und. Was passiert während einer Amazon ECS-Bereitstellung Weitere Informationen zu Aufgabensätzen finden Sie TaskSet in der Amazon Elastic Container Service API-Referenz und in describe-task-set im Abschnitt Amazon ECS der Befehlsreferenz. AWS CLI

Mögliche Lösung: Stellen Sie sicher, dass sowohl der Produktions-Listener als auch der Test-Listener von Elastic Load Balancing auf die Zielgruppe verweisen, die derzeit Ihre Workloads bedient. Es gibt drei Stellen, die Sie überprüfen sollten:

- In Amazon EC2 in den Listener- und Regeleinstellungen Ihres Load Balancers. Weitere Informationen finden Sie unter Listener für Ihre Application Load Balancer im Benutzerhandbuch für Application Load Balancer oder Listener für Ihre Network Load Balancer im Benutzerhandbuch für Network Load Balancer.
- In Amazon ECS, in Ihrem Cluster, unter der Netzwerkkonfiguration Ihres Dienstes. Weitere Informationen finden Sie unter Überlegungen zu Application Load Balancer und Network Load Balancer im Amazon Elastic Container Service Developer Guide.
- In CodeDeploy, in den Einstellungen Ihrer Bereitstellungsgruppe. Weitere Informationen finden Sie unter Eine Bereitstellungsgruppe für eine Amazon ECS-Bereitstellung erstellen (Konsole).

Meine Bereitstellung schlägt manchmal fehl, wenn Auto Scaling verwendet wird

Problem: Sie verwenden Auto Scaling mit CodeDeploy und stellen fest, dass Ihre Bereitstellungen gelegentlich fehlschlagen. Weitere Informationen zu den Symptomen dieses Problems finden Sie im Amazon Elastic Container Service Developer Guide unter dem Thema Für Dienste, die für die Verwendung von Service Auto Scaling konfiguriert sind, und den blue/green Bereitstellungstyp, wird Auto Scaling während einer Bereitstellung nicht blockiert, aber die Bereitstellung kann unter bestimmten Umständen fehlschlagen.

Mögliche Ursache: Dieses Problem kann auftreten, wenn CodeDeploy die Auto Scaling-Prozesse miteinander in Konflikt geraten.

Mögliche Lösung: Unterbrechen und setzen Sie Auto Scaling-Prozesse während der CodeDeploy Bereitstellung mithilfe der `RegisterScalableTarget` API (oder des entsprechenden `register-scalable-target` AWS CLI Befehls) fort. Weitere Informationen finden Sie unter Unterbrechen und Fortsetzen der Skalierung für Application Auto Scaling im Application Auto Scaling-Benutzerhandbuch.

> **Note**
>
> CodeDeploy kann nicht `RegisterScaleableTarget` direkt anrufen. Um diese API verwenden zu können, müssen Sie so konfigurieren, CodeDeploy dass eine Benachrichtigung oder ein Ereignis an Amazon Simple Notification Service (oder Amazon CloudWatch) gesendet wird. Anschließend müssen Sie Amazon SNS (oder CloudWatch) so konfigurieren, dass eine Lambda-Funktion aufgerufen wird, und die Lambda-Funktion so konfigurieren, dass sie die API aufruft. `RegisterScalableTarget` Die `RegisterScalableTarget` API muss mit dem `SuspendedState` Parameter auf auf aufgerufen werden, `true` um Auto Scaling-Operationen auszusetzen und wieder `false` aufzunehmen.
>
> Die Benachrichtigung oder das Ereignis, das CodeDeploy gesendet wird, muss eintreten, wenn eine Bereitstellung gestartet wird (um Auto Scaling-Sperrvorgänge auszulösen) oder wenn eine Bereitstellung erfolgreich ist, fehlschlägt oder beendet wird (um Auto Scaling-Wiederaufnahmevorgänge auszulösen).
>
> Informationen zur Konfiguration CodeDeploy der Generierung von Amazon SNS-Benachrichtigungen oder CloudWatch -Ereignissen finden Sie unterÜberwachung von

Bereitstellungen mit Amazon Events CloudWatch. und. Monitoring Deployments with Amazon SNS Event Notifications

Nur ALB unterstützt das schrittweise Routing des Datenverkehrs. Verwenden Sie stattdessen das AllAtOnce Traffic-Routing, wenn Sie create/update eine Gruppe einrichten

Problem: Beim Erstellen oder Aktualisieren einer Bereitstellungsgruppe wird die folgende Fehlermeldung angezeigt in CodeDeploy:

```
Only ALB supports gradual traffic routing, use AllAtOnce Traffic routing instead when you create/update Deployment group.
```

Mögliche Ursache: Dieser Fehler kann auftreten, wenn Sie einen Network Load Balancer verwenden und versuchen, eine andere vordefinierte Bereitstellungskonfiguration zu verwenden. `CodeDeployDefault.ECSAllAtOnce`

Mögliche Lösungen:

- Ändern Sie Ihre vordefinierte Bereitstellungskonfiguration in`CodeDeployDefault.ECSAllAtOnce`. Dies ist die einzige vordefinierte Bereitstellungskonfiguration, die von Network Load Balancers unterstützt wird.

  Weitere Informationen zu vordefinierten Bereitstellungskonfigurationen finden Sie unterVordefinierte Bereitstellungskonfigurationen für eine Amazon ECS-Rechenplattform.
- Ändern Sie Ihren Load Balancer in einen Application Load Balancer. Die Application Load Balancer unterstützen alle vordefinierten Bereitstellungskonfigurationen. Weitere Informationen zum Erstellen eines Application Load Balancers finden Sie unter. Einen Load Balancer, Zielgruppen und Listener für CodeDeploy Amazon ECS-Bereitstellungen einrichten

Obwohl meine Bereitstellung erfolgreich war, besteht der Ersatz-Tasksatz bei den Elastic Load Balancing-Zustandsprüfungen nicht und meine Anwendung ist ausgefallen

Problem: Obwohl es CodeDeploy darauf hinweist, dass meine Bereitstellung erfolgreich war, besteht das Ersatz-Task-Set die Zustandsprüfungen von Elastic Load Balancing nicht und meine Anwendung ist ausgefallen.

Mögliche Ursache: Dieses Problem kann auftreten, wenn Sie eine CodeDeploy Komplettbereitstellung durchgeführt haben und Ihr Ersatz-Task-Set (grün) fehlerhaften Code enthält, der dazu führt, dass die Elastic Load Balancing-Zustandsprüfungen fehlschlagen. Bei der All-at-Once-Bereitstellungskonfiguration werden die Zustandsprüfungen des Load Balancers für den Ersatz-Tasksatz ausgeführt, nachdem der Datenverkehr dorthin verlagert wurde (d. h., wenn das Lifecycle-Ereignis danach CodeDeploy eingetreten ist). `AllowTraffic` Aus diesem Grund werden Sie feststellen, dass die Systemdiagnosen für den Ersatz-Tasksatz nach einer Verlagerung des Datenverkehrs fehlschlagen, aber nicht davor. Informationen zu den CodeDeploy generierten Lebenszyklusereignissen finden Sie unterWas passiert während einer Amazon ECS-Bereitstellung.

Mögliche Lösungen:

- Ändern Sie Ihre Bereitstellungskonfiguration von All-at-Once auf Canary oder Linear. In einer kanarischen oder linearen Konfiguration beginnen die Zustandsprüfungen des Load Balancers für die Ersatzaufgabe, während Ihre Anwendung in der Ersatzumgebung CodeDeploy installiert wird und bevor der Datenverkehr verschoben wird (d. h. während des `Install` Lebenszyklusereignisses und vor dem Ereignis). `AllowTraffic` Wenn Sie zulassen, dass die Prüfungen während der Anwendungsinstallation, aber bevor der Datenverkehr verlagert wird, ausgeführt werden, wird fehlerhafter Anwendungscode erkannt, der zu Bereitstellungsfehlern führt, bevor die Anwendung öffentlich verfügbar ist.

  Hinweise zur Konfiguration von Canary- oder Linear-Bereitstellungen finden Sie unterÄndern Sie die Einstellungen der Bereitstellungsgruppe mit CodeDeploy.

  Informationen zu CodeDeploy Lebenszyklusereignissen, die während einer Amazon ECS-Bereitstellung auftreten, finden Sie unterWas passiert während einer Amazon ECS-Bereitstellung.

> **Note**
>
> Kanarische und lineare Bereitstellungskonfigurationen werden nur mit Application Load Balancers unterstützt.

- Wenn Sie Ihre All-at-Once-Bereitstellungskonfiguration beibehalten möchten, richten Sie einen Test-Listener ein und überprüfen Sie den Status des Ersatz-Task-Sets mit dem Lifecycle-Hook. `BeforeAllowTraffic` Weitere Informationen finden Sie unter Liste der Lifecycle-Event-Hooks für eine Amazon ECS-Bereitstellung.

Kann ich einer Bereitstellungsgruppe mehrere Load Balancer zuordnen?

Nein. Wenn Sie mehrere Application Load Balancer oder Network Load Balancer verwenden möchten, verwenden Sie Amazon ECS Rolling Updates statt CodeDeploy blue/green Bereitstellungen. Weitere Informationen zu fortlaufenden Updates finden Sie unter Rolling Update im Amazon Elastic Container Service Developer Guide. Weitere Informationen zur Verwendung mehrerer Load Balancer mit Amazon ECS finden Sie unter Registrieren mehrerer Zielgruppen bei einem Service im Amazon Elastic Container Service Developer Guide.

Kann ich CodeDeploy blue/green Bereitstellungen ohne einen Load Balancer durchführen?

Nein, ohne einen Load Balancer können Sie keine CodeDeploy blue/green Bereitstellungen durchführen. Wenn Sie keinen Load Balancer verwenden können, verwenden Sie stattdessen die Funktion für fortlaufende Updates von Amazon ECS. Weitere Informationen zu fortlaufenden Amazon ECS-Updates finden Sie unter Rolling Update im Amazon Elastic Container Service Developer Guide.

Wie kann ich meinen Amazon ECS-Service während einer Bereitstellung mit neuen Informationen aktualisieren?

Damit Ihr Amazon ECS-Service während der Bereitstellung mit einem neuen Parameter CodeDeploy aktualisiert wird, geben Sie den Parameter im `resources` Abschnitt der AppSpec Datei an. Nur wenige Amazon ECS-Parameter werden von unterstützt CodeDeploy, z. B. die Aufgabendefinitionsdatei und die Parameter für den Containernamen. Eine vollständige Liste der

Amazon ECS-Parameter, die aktualisiert CodeDeploy werden können, finden Sie unterAppSpec Abschnitt „Ressourcen“ für Amazon ECS-Bereitstellungen.

> **Note**
>
> Wenn Sie Ihren Amazon ECS-Service mit einem Parameter aktualisieren müssen, der von nicht unterstützt wird CodeDeploy, gehen Sie wie folgt vor:
>
> 1. Rufen Sie die Amazon `UpdateService` ECS-API mit dem Parameter auf, den Sie aktualisieren möchten. Eine vollständige Liste der Parameter, die aktualisiert werden können, finden Sie UpdateService in der Amazon Elastic Container Service API-Referenz.
> 2. Um die Änderung auf die Aufgaben anzuwenden, erstellen Sie eine neue Amazon blue/green ECS-Bereitstellung. Weitere Informationen finden Sie unter Erstellen Sie eine Amazon ECS Compute Platform-Bereitstellung (Konsole).

Fehlerbehebung AWS Probleme bei der Lambda-Bereitstellung

Themen

- AWS Lambda Bereitstellungen schlagen fehl, nachdem eine Lambda-Bereitstellung, für die keine Rollbacks konfiguriert wurden, manuell gestoppt wurde

AWS Lambda Bereitstellungen schlagen fehl, nachdem eine Lambda-Bereitstellung, für die keine Rollbacks konfiguriert wurden, manuell gestoppt wurde

In einigen Fällen kann der Alias einer in einer Bereitstellung angegebenen Lambda-Funktion auf zwei verschiedene Versionen der Funktion verweisen. Das Ergebnis ist, dass nachfolgende Versuche, die Lambda-Funktion bereitzustellen, fehlschlagen. Eine Lambda-Bereitstellung kann in diesen Zustand gelangen, wenn keine Rollbacks konfiguriert sind und sie manuell gestoppt wird. Um fortzufahren, stellen Sie mithilfe der AWS Lambda Konsole sicher, dass die Funktion nicht so konfiguriert ist, dass sie den Datenverkehr zwischen zwei Versionen verschiebt:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.

2. Wählen Sie im linken Bereich Functions (Funktionen) aus.
3. Wählen Sie den Namen der Lambda-Funktion aus, die sich in Ihrer CodeDeploy Bereitstellung befindet.
4. Wählen Sie unter Aliase den Alias aus, der in Ihrer CodeDeploy Bereitstellung verwendet wurde, und klicken Sie dann auf Bearbeiten.
5. Wählen Sie im Feld Gewichteter Alias die Option aus**none**. Dadurch wird sichergestellt, dass der Alias nicht so konfiguriert ist, dass es einen Prozentsatz oder eine Gewichtung des Datenverkehrs auf mehr als eine Version verschiebt. Notieren Sie die in Version ausgewählte Version.
6. Wählen Sie Speichern.
7. Öffnen Sie die CodeDeploy Konsole und versuchen Sie, die Version bereitzustellen, die im Dropdownmenü in Schritt 5 angezeigt wird.

Beheben von Problemen bei Bereitstellungsgruppen

Durch Markieren einer Instance als Teil einer Bereitstellungsgruppe wird Ihre Anwendung nicht automatisch in der neuen Instance bereitgestellt

CodeDeploy stellt Ihre Anwendung nicht automatisch auf einer neu markierten Instanz bereit. Sie müssen eine neue Bereitstellung in der Bereitstellungsgruppe erstellen.

Sie können CodeDeploy es verwenden, um automatische Bereitstellungen für neue EC2-Instances in Amazon EC2 Auto Scaling Scaling-Gruppen zu aktivieren. Weitere Informationen finden Sie unter Integration CodeDeploy mit Amazon EC2 Auto Scaling.

Beheben von Instance-Problemen

Themen

- Tags müssen korrekt festgelegt sein
- AWS CodeDeploy Der Agent muss auf Instanzen installiert sein und ausgeführt werden
- Bereitstellungen schlagen für bis zu einer Stunde nicht fehl, nachdem eine Instance während einer Bereitstellung beendet wurde
- Analysieren von Protokolldateien zur Untersuchung von Bereitstellungsfehlern auf Instances
- Erstellen Sie eine neue CodeDeploy Protokolldatei, falls sie versehentlich gelöscht wurde

- Problembehandlung bei Bereitstellungsfehlern „InvalidSignatureException — Signatur ist abgelaufen: [Zeit] ist jetzt vor [Zeit]“

Tags müssen korrekt festgelegt sein

Verwenden Sie den Befehl list-deployment-instances, um zu überprüfen, ob die für eine Bereitstellung verwendeten Instances korrekt gekennzeichnet sind. Wenn eine EC2-Instance in der Ausgabe fehlt, überprüfen Sie mithilfe der EC2-Konsole, ob die Tags für die Instance gesetzt wurden. Weitere Informationen finden Sie unter Arbeiten mit Tags in der Konsole im Amazon EC2 EC2-Benutzerhandbuch.

> **Note**
>
> Wenn Sie eine Instance taggen und sofort eine Anwendung für sie bereitstellen, ist die Instance möglicherweise nicht in der Bereitstellung enthalten. CodeDeploy Das liegt daran, dass es mehrere Minuten dauern CodeDeploy kann, bis die Tags gelesen werden können. Wir empfehlen, dass Sie mindestens fünf Minuten zwischen dem Zeitpunkt, zu dem Sie eine Instance taggen, und dem Versuch einer Bereitstellung warten.

AWS CodeDeploy Der Agent muss auf Instanzen installiert sein und ausgeführt werden

Informationen dazu, wie Sie überprüfen können, ob der CodeDeploy Agent auf einer Instanz installiert ist und ausgeführt wird, finden Sie unterStellen Sie sicher, dass der CodeDeploy Agent läuft.

Informationen zur Installation, Deinstallation oder Neuinstallation des CodeDeploy Agenten finden Sie unterInstallieren Sie den CodeDeploy Agenten.

Bereitstellungen schlagen für bis zu einer Stunde nicht fehl, nachdem eine Instance während einer Bereitstellung beendet wurde

CodeDeploy bietet ein Zeitfenster von einer Stunde, in dem jedes Ereignis im Bereitstellungszyklus bis zum Abschluss ausgeführt werden kann. Dies bietet ausreichend Zeit für lange andauernde Skripts.

Wenn die Skripts während eines laufenden Lebenszyklusereignisses nicht vollständig ausgeführt werden (z. B. wenn eine Instanz beendet oder der CodeDeploy Agent heruntergefahren wird), kann

es bis zu einer Stunde dauern, bis der Status der Bereitstellung als Fehlgeschlagen angezeigt wird. Dies gilt auch, wenn der im Skript angegebene Timeout-Zeitraum weniger als eine Stunde beträgt. Das liegt daran, dass der CodeDeploy Agent heruntergefahren wird, wenn die Instanz beendet wird und keine weiteren Skripts mehr verarbeiten kann.

Wenn eine Instance zwischen Lebenszyklusereignissen oder vor dem ersten Schritt des Lebenszyklusereignisses beendet wird, tritt die Zeitüberschreitung bereits nach fünf Minuten ein.

Analysieren von Protokolldateien zur Untersuchung von Bereitstellungsfehlern auf Instances

Wenn der Status einer Instance in der Bereitstellung anders als `Succeeded` lautet, finden Sie in der Protokolldatei der Bereitstellung Daten zur Identifizierung des Problems. Weitere Informationen über den Zugriff auf Bereitstellungsprotokolldaten finden Sie unter Protokolldaten für CodeDeploy EC2/On-Premises Bereitstellungen anzeigen.

Erstellen Sie eine neue CodeDeploy Protokolldatei, falls sie versehentlich gelöscht wurde

Wenn Sie versehentlich die Bereitstellungsprotokolldatei auf einer Instanz löschen, CodeDeploy wird keine Ersatzprotokolldatei erstellt. Zum Erstellen einer neuen Protokolldatei melden Sie sich bei der Instance an und führen dann die folgenden Befehle aus:

Führen Sie für eine Amazon Linux-, Ubuntu Server- oder RHEL-Instance die folgenden Befehle nacheinander in dieser Reihenfolge aus:

```
systemctl stop codedeploy-agent
```

```
systemctl start codedeploy-agent
```

Für eine Windows Server-Instance:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Problembehandlung bei Bereitstellungsfehlern „InvalidSignatureException — Signatur ist abgelaufen: [Zeit] ist jetzt vor [Zeit]“

CodeDeploy benötigt für die Ausführung seiner Operationen genaue Zeitangaben. Wenn Datum und Uhrzeit auf Ihrer Instance nicht korrekt eingestellt sind, stimmen sie möglicherweise nicht mit dem Signaturdatum Ihrer Bereitstellungsanfrage überein, die CodeDeploy abgelehnt wird.

Informationen dazu, wie Sie Bereitstellungsfehler aufgrund falscher Zeiteinstellungen vermeiden, finden Sie in den folgenden Themen:

- Einstellen der Zeit einer Linux-Instance
- Einstellen der Zeit für eine Windows-Instance

Beheben Sie GitHub Token-Probleme

Ungültiges GitHub OAuth-Token

CodeDeploy Anwendungen, die nach Juni 2017 erstellt wurden, verwenden GitHub OAuth-Token für jede Region. AWS Durch die Verwendung von Token, die an bestimmte AWS Regionen gebunden sind, haben Sie mehr Kontrolle darüber, welche CodeDeploy Anwendungen Zugriff auf ein GitHub Repository haben.

Wenn Sie einen GitHub Token-Fehler erhalten, haben Sie möglicherweise ein älteres Token, das jetzt ungültig ist.

Um ein ungültiges GitHub OAuth-Token zu korrigieren

1. Entfernen Sie das alte Token mit einer der folgenden Methoden:
   - Um das alte Token mithilfe der API zu entfernen, verwenden Sie DeleteGitHubAccountToken.
   - Um das alte Token zu entfernen, verwenden Sie AWS Command Line Interface:
     a. Gehen Sie zu dem Computer, auf dem sich das Token befindet.
     b. Stellen Sie sicher, dass der auf diesem Computer installiert AWS CLI ist. Installationsanweisungen finden Sie unter Installation, Aktualisierung und Deinstallation von AWS CLI im AWS Command Line Interface Benutzerhandbuch

c. Geben Sie auf dem Computer, auf dem sich das Token befindet, den folgenden Befehl ein:

```
aws delete-git-hub-account-token
```

Einzelheiten zur Befehlssyntax finden Sie unter delete-git-hub-account-token.

2. Fügen Sie ein neues OAuth-Token hinzu. Weitere Informationen finden Sie unter Integrieren CodeDeploy mit GitHub.

GitHub Die maximale Anzahl von OAuth-Token wurde überschritten

Wenn Sie eine CodeDeploy Bereitstellung erstellen, beträgt die maximale Anzahl zulässiger GitHub Token 10. Wenn Sie eine Fehlermeldung zu GitHub OAuth-Token erhalten, stellen Sie sicher, dass Sie über 10 oder weniger Token verfügen. Wenn Sie mehr als 10 Token haben, sind die ersten Token, die erstellt wurden, ungültig. Wenn Sie beispielsweise 11 Token haben, ist das erste Token, das Sie erstellt haben, ungültig. Wenn Sie 12 Token haben, sind die ersten beiden Token, die Sie erstellt haben, ungültig. Informationen zur Verwendung der CodeDeploy API zum Entfernen alter Token finden Sie unter. DeleteGitHubAccountToken

Probleme mit Amazon EC2 Auto Scaling beheben

Themen

- Allgemeine Amazon EC2 Auto Scaling Scaling-Fehlerbehebung
- "gibt Ihnen CodeDeployRole nicht die Erlaubnis, folgende Operationen auszuführen AWS Service: AmazonAutoScaling "Fehler
- Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe werden kontinuierlich bereitgestellt und beendet, bevor eine Revision bereitgestellt werden kann
- Das Beenden oder Neustarten einer Amazon EC2 Auto Scaling Scaling-Instance kann dazu führen, dass Bereitstellungen fehlschlagen
- Vermeiden Sie es, mehrere Bereitstellungsgruppen einer einzigen Amazon EC2 Auto Scaling-Gruppe zuzuordnen
- EC2-Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe können nicht gestartet werden und erhalten den Fehler „Heartbeat Timeout“

- Nicht übereinstimmende Amazon EC2 Auto Scaling Scaling-Lifecycle-Hooks können dazu führen, dass automatische Bereitstellungen für Amazon EC2 Auto Scaling Scaling-Gruppen gestoppt werden oder fehlschlagen
- Fehler „Die Bereitstellung ist fehlgeschlagen, weil keine Instances für Ihre Bereitstellungsgruppe gefunden wurden“

Allgemeine Amazon EC2 Auto Scaling Scaling-Fehlerbehebung

Bereitstellungen für EC2-Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe können aus den folgenden Gründen fehlschlagen:

- Amazon EC2 Auto Scaling startet und beendet kontinuierlich EC2-Instances. Wenn Ihre Anwendungsversion CodeDeploy nicht automatisch bereitgestellt werden kann, startet und beendet Amazon EC2 Auto Scaling kontinuierlich EC2-Instances.

  Trennen Sie die Amazon EC2 Auto Scaling-Gruppe von der CodeDeploy Bereitstellungsgruppe oder ändern Sie die Konfiguration Ihrer Amazon EC2 Auto Scaling Scaling-Gruppe so, dass die gewünschte Anzahl von Instances der aktuellen Anzahl von Instances entspricht (wodurch verhindert wird, dass Amazon EC2 Auto Scaling weitere EC2-Instances startet). Weitere Informationen finden Sie unter Ändern Sie die Einstellungen der Bereitstellungsgruppe mit CodeDeploy oder Manuelle Skalierung für Amazon EC2 Auto Scaling.
- Der CodeDeploy Agent reagiert nicht. Der CodeDeploy Agent wird möglicherweise nicht installiert, wenn Initialisierungsskripts (z. B. Cloud-Init-Skripts), die unmittelbar nach dem Starten oder Starten einer EC2-Instance ausgeführt werden, mehr als eine Stunde benötigen. CodeDeploy hat ein Timeout von einer Stunde, bis der CodeDeploy Agent auf ausstehende Bereitstellungen reagiert. Um dieses Problem zu beheben, verschieben Sie Ihre Initialisierungsskripts in Ihre CodeDeploy-Anwendungsrevision.
- Eine EC2-Instance in einer Amazon EC2 Auto Scaling Scaling-Gruppe wird während einer Bereitstellung neu gestartet. Ihre Bereitstellung kann fehlschlagen, wenn eine EC2-Instance während einer Bereitstellung neu gestartet wird oder der CodeDeploy Agent während der Verarbeitung eines Bereitstellungsbefehls heruntergefahren wird. Weitere Informationen finden Sie unter Das Beenden oder Neustarten einer Amazon EC2 Auto Scaling Scaling-Instance kann dazu führen, dass Bereitstellungen fehlschlagen.
- Mehrere Anwendungsrevisionen werden gleichzeitig auf derselben EC2-Instance in einer Amazon EC2 Auto Scaling Scaling-Gruppe bereitgestellt. Die gleichzeitige Bereitstellung mehrerer Anwendungsrevisionen auf derselben EC2-Instance in einer Amazon EC2 Auto Scaling Scaling-

Gruppe kann fehlschlagen, wenn eine der Bereitstellungen Skripts enthält, die länger als ein paar Minuten ausgeführt werden. Stellen Sie nicht mehrere Anwendungsrevisionen für dieselben EC2-Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe bereit.

- Eine Bereitstellung schlägt für neue EC2-Instances fehl, die als Teil einer Amazon EC2 Auto Scaling Scaling-Gruppe gestartet werden. In diesem Szenario kann die Ausführung der Skripts in einer Bereitstellung den Start von EC2-Instances in der Amazon EC2 Auto Scaling Scaling-Gruppe verhindern. (Andere EC2-Instances in der Amazon EC2 Auto Scaling Scaling-Gruppe scheinen normal zu laufen.) Um dieses Problem zu beheben, stellen Sie sicher, dass alle anderen Skripts abgeschlossen sind:
  - CodeDeploy Agent ist nicht in Ihrem AMI enthalten: Wenn Sie den cfn-init Befehl verwenden, um den CodeDeploy Agenten zu installieren, während Sie eine neue Instance starten, platzieren Sie das Agent-Installationsskript am Ende des `cfn-init` Abschnitts Ihrer CloudFormation Vorlage.
  - CodeDeploy Agent ist in Ihrem AMI enthalten: Konfigurieren Sie das AMI so, dass sich der Agent bei der Erstellung der Instance in einem `Stopped` Zustand befindet, und fügen Sie dann als letzten Schritt ein Skript zum Starten des Agenten in Ihre `cfn-init` Skriptbibliothek ein.

"gibt Ihnen CodeDeployRole nicht die Erlaubnis, folgende Operationen auszuführen AWS Service: AmazonAutoScaling "Fehler

Bereitstellungen, die eine mit einer Startvorlage erstellte Auto Scaling Scaling-Gruppe verwenden, benötigen die folgenden Berechtigungen. Diese gelten zusätzlich zu den Berechtigungen, die durch die `AWSCodeDeployRole` AWS verwaltete Richtlinie gewährt werden.

- `EC2:RunInstances`
- `EC2:CreateTags`
- `iam:PassRole`

Dieser Fehler wird möglicherweise angezeigt, wenn Ihnen diese Berechtigungen fehlen. Weitere Informationen finden Sie unter Tutorial: Wird verwendet CodeDeploy , um eine Anwendung für eine Auto Scaling Scaling-Gruppe bereitzustellen Erstellen einer Startvorlage für eine Auto Scaling Scaling-Gruppe und Berechtigungen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe werden kontinuierlich bereitgestellt und beendet, bevor eine Revision bereitgestellt werden kann

In einigen Fällen kann ein Fehler eine erfolgreiche Bereitstellung für neu bereitgestellte Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe verhindern. Dies führt zu keinen fehlerfreien Instances und keinen erfolgreichen Bereitstellungen. Da die Bereitstellung nicht erfolgreich ausgeführt oder nicht abgeschlossen werden kann, werden die Instances bald nach der Erstellung beendet. Die Amazon EC2 Auto Scaling Scaling-Gruppenkonfiguration veranlasst dann, dass ein weiterer Batch von Instances bereitgestellt wird, um zu versuchen, die Mindestanforderung für gesunde Hosts zu erfüllen. Dieser Stapel wird außerdem beendet und der Zyklus wird fortgesetzt.

Mögliche Gründe hierfür sind:

- Fehlgeschlagene Zustandsprüfungen der Amazon EC2 Auto Scaling Scaling-Gruppe.
- Ein Fehler in der Anwendungsrevision

Um dieses Problem zu umgehen, führen Sie die folgenden Schritte aus:

1. Erstellen Sie manuell eine EC2-Instance, die nicht Teil der Amazon EC2 Auto Scaling Scaling-Gruppe ist. Markieren Sie die Instance mit einem eindeutigen EC2-Instance-Tag.
2. Fügen Sie die neue Instance zu der betroffenen Bereitstellungsgruppe hinzu.
3. Stellen Sie eine neue, fehlerfreie Anwendungsrevision für die Bereitstellungsgruppe bereit.

Dadurch wird die Amazon EC2 Auto Scaling Scaling-Gruppe aufgefordert, die Anwendungsrevision für future Instances in der Amazon EC2 Auto Scaling Scaling-Gruppe bereitzustellen.

> **Note**
>
> Nachdem Sie bestätigt haben, dass die Bereitstellungen erfolgreich waren, löschen Sie die Instance, die Sie erstellt haben, um laufende Gebühren für Ihr Konto zu vermeiden. AWS

Das Beenden oder Neustarten einer Amazon EC2 Auto Scaling Scaling-Instance kann dazu führen, dass Bereitstellungen fehlschlagen

Wenn eine EC2-Instance über Amazon EC2 Auto Scaling gestartet wird und die Instance anschließend beendet oder neu gestartet wird, können Bereitstellungen für diese Instance aus den folgenden Gründen fehlschlagen:

- Während einer laufenden Bereitstellung führt ein Scale-In-Ereignis oder ein anderes Kündigungsereignis dazu, dass sich die Instance von der Amazon EC2 Auto Scaling Scaling-Gruppe trennt und dann beendet wird. Da die Bereitstellung nicht abgeschlossen werden kann, schlägt sie fehl.
- Die Instance wird neu gestartet, aber es dauert mehr als fünf Minuten, bis die Instance gestartet wird. CodeDeploy behandelt dies als Timeout. Der Service lässt alle aktuellen und zukünftigen Bereitstellungen in der Instance fehlschlagen.

So beheben Sie dieses Problem:

- Stellen Sie grundsätzlich sicher, dass alle Bereitstellungen abgeschlossen sind, bevor die Instance beendet oder neu gestartet wird. Stellen Sie sicher, dass alle Bereitstellungen starten, nachdem die Instance gestartet oder neu gestartet wurde.
- Bereitstellungen können fehlschlagen, wenn Sie ein Amazon Machine Image (AMI) auf Windows Server-Basis für eine Amazon EC2 Auto Scaling Scaling-Konfiguration angeben und den EC2Config-Service verwenden, um den Computernamen der Instance festzulegen. Um dieses Problem zu beheben, deaktivieren Sie im Windows Server-Basis-AMI auf der Registerkarte Allgemein in den EC2-Diensteigenschaften die Option Computername festlegen. Nachdem Sie dieses Kontrollkästchen deaktiviert haben, ist dieses Verhalten für alle neuen Windows Server Amazon EC2 Auto Scaling Scaling-Instances deaktiviert, die mit diesem Windows Server-Basis-AMI gestartet wurden. Für Windows Server Amazon EC2 Auto Scaling Scaling-Instances, auf denen dieses Verhalten aktiviert ist, müssen Sie dieses Kontrollkästchen nicht deaktivieren. Stellen Sie fehlgeschlagene Bereitstellungen einfach erneut in diesen Instances bereit, nachdem sie neu gestartet wurden.

Vermeiden Sie es, mehrere Bereitstellungsgruppen einer einzigen Amazon EC2 Auto Scaling-Gruppe zuzuordnen

Als bewährte Methode sollten Sie jeder Amazon EC2 Auto Scaling Scaling-Gruppe nur eine Bereitstellungsgruppe zuordnen.

Das liegt daran, dass Amazon EC2 Auto Scaling, wenn es eine Instance hochskaliert, deren Hooks mehreren Bereitstellungsgruppen zugeordnet sind, Benachrichtigungen für alle Hooks gleichzeitig sendet. Dies bewirkt, dass mehrere Bereitstellungen für jede Instance gleichzeitig starten. Wenn mehrere Bereitstellungen gleichzeitig Befehle an den CodeDeploy Agenten senden, kann das Zeitlimit von fünf Minuten zwischen einem Lebenszyklusereignis und entweder dem Beginn der Bereitstellung oder dem Ende des vorherigen Lebenszyklusereignisses erreicht sein. In diesem Fall schlägt die Bereitstellung fehl, auch wenn ein Bereitstellungsprozess ansonsten wie erwartet abläuft.

> **Note**
>
> Das Standard-Timeout für ein Skript in einem Lebenszyklusereignis beträgt 30 Minuten. Sie können das Timeout in der Datei auf einen anderen Wert ändern. AppSpec Weitere Informationen finden Sie unter Fügen Sie eine AppSpec Datei für eine EC2/On-Premises Bereitstellung hinzu.

Es ist nicht möglich, die Reihenfolge der Bereitstellungen zu steuern, wenn mehrere Bereitstellungen gleichzeitig ausgeführt werden sollen.

Und wenn die Bereitstellung auf einer Instance fehlschlägt, beendet Amazon EC2 Auto Scaling die Instance sofort. Wenn die erste Instance heruntergefahren wird, beginnen die anderen laufenden Bereitstellungen auszufallen. Da es für den CodeDeploy Agenten ein Timeout von einer Stunde CodeDeploy gibt, um auf ausstehende Bereitstellungen zu reagieren, kann es bis zu 60 Minuten dauern, bis das Timeout für jede Instance eintritt.

Weitere Informationen zu Amazon EC2 Auto Scaling finden Sie unter Under the hood: CodeDeploy and Auto Scaling-Integration.

EC2-Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe können nicht gestartet werden und erhalten den Fehler „Heartbeat Timeout“

Eine Amazon EC2 Auto Scaling Scaling-Gruppe kann möglicherweise keine neuen EC2-Instances starten und generiert eine Meldung ähnlich der folgenden:

`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.`

Diese Meldung weist normalerweise auf einen der folgenden Sachverhalte hin:

- Die maximale Anzahl gleichzeitiger Bereitstellungen, die einem AWS Konto zugeordnet sind, wurde erreicht. Weitere Informationen zu Bereitstellungslimits finden Sie unter CodeDeploy Kontingente.
- Die Auto Scaling Scaling-Gruppe hat versucht, zu viele EC2-Instances zu schnell zu starten. Die API-Aufrufe zu RecordLifecycleActionHeartbeatoder CompleteLifecycleActionfür jede neue Instance wurden gedrosselt.
- Eine Anwendung CodeDeploy wurde gelöscht, bevor die zugehörigen Bereitstellungsgruppen aktualisiert oder gelöscht wurden.

  Wenn Sie eine Anwendung oder Bereitstellungsgruppe löschen, CodeDeploy versucht es, alle damit verknüpften Amazon EC2 Auto Scaling Scaling-Hooks zu bereinigen, aber einige Hooks bleiben möglicherweise bestehen. Wenn Sie einen Befehl zum Löschen einer Bereitstellungsgruppe ausführen, werden die restlichen Hooks in der Ausgabe zurückgegeben. Wenn Sie jedoch einen Befehl zum Löschen einer Anwendung ausführen, werden die restlichen Hooks nicht in der Ausgabe angezeigt.

  Löschen Sie daher als bewährte Methode alle Bereitstellungsgruppen, die einer Anwendung zugeordnet sind, bevor Sie die Anwendung löschen. Sie können anhand der Befehlsausgabe die Lebenszyklus-Hooks bestimmen, die manuell gelöscht werden müssen.

Wenn Sie die Fehlermeldung „Heartbeat Timeout“ erhalten, können Sie feststellen, ob übrig gebliebene Lifecycle-Hooks die Ursache sind, und das Problem lösen, indem Sie wie folgt vorgehen:

1. Führen Sie eine der folgenden Aktionen aus:

   - Rufen Sie den Befehl delete-deployment-group auf, um die Bereitstellungsgruppe zu löschen, die der Auto Scaling Scaling-Gruppe zugeordnet ist, die das Heartbeat-Timeout verursacht.
   - Rufen Sie den Befehl update-deployment-group mit einer leeren Liste von Auto Scaling-Gruppennamen auf, um alle Auto Scaling-Lifecycle-Hooks zu trennen. CodeDeploy-managed

     Geben Sie beispielsweise den folgenden Befehl ein: AWS CLI

```
aws deploy update-deployment-group --application-name my-example-app
--current-deployment-group-name my-deployment-group --auto-scaling-
groups
```

Ein weiteres Beispiel: Wenn Sie die CodeDeploy API mit Java verwenden, rufen Sie auf `UpdateDeploymentGroup` und setzen Sie `autoScalingGroups` auf`new ArrayList<String>()`. Dadurch wird `autoScalingGroups` eine leere Liste angezeigt und die bestehende Liste wird entfernt. Verwenden Sie diese Option nicht`null`, was die Standardeinstellung ist, da sie so `autoScalingGroups` bleibt, wie sie ist, was Sie nicht möchten.

Überprüfen Sie die Ausgabe des Aufrufs. Wenn die Ausgabe eine `hooksNotCleanedUp` Struktur mit einer Liste von Amazon EC2 Auto Scaling Scaling-Lifecycle-Hooks enthält, gibt es übrig gebliebene Lifecycle-Hooks.

2. Rufen Sie den Befehl describe-lifelycle-hooks auf und geben Sie den Namen der Amazon EC2 Auto Scaling Scaling-Gruppe an, die den EC2-Instances zugeordnet ist, die nicht gestartet werden konnten. Suchen Sie in der Ausgabe nach einem der folgenden Elemente:

   - Amazon EC2 Auto Scaling Scaling-Lifecycle-Hook-Namen, die der `hooksNotCleanedUp` Struktur entsprechen, die Sie in Schritt 1 identifiziert haben.
   - Amazon EC2 Auto Scaling Scaling-Lifecycle-Hook-Namen, die den Namen der Bereitstellungsgruppe enthalten, die der fehlerhaften Auto Scaling Scaling-Gruppe zugeordnet ist.
   - Namen von Amazon EC2 Auto Scaling Scaling-Lifecycle-Hooks, die möglicherweise das Heartbeat-Timeout für die Bereitstellung verursacht haben. CodeDeploy

3. Wenn ein Hook in eine der in Schritt 2 aufgeführten Kategorien fällt, rufen Sie den Befehl delete-lifecycle ycle-hook auf, um ihn zu löschen. Geben Sie die Amazon EC2 Auto Scaling Scaling-Gruppe und den Lifecycle-Hook im Aufruf an.

> **Important**
>
> Löschen Sie nur Hooks, die Probleme verursachen, wie in Schritt 2 beschrieben. Wenn Sie brauchbare Hooks löschen, schlagen Ihre Bereitstellungen möglicherweise fehl oder Sie sind CodeDeploy möglicherweise nicht in der Lage, Ihre Anwendungsrevisionen auf skalierten EC2-Instances bereitzustellen.

4. Rufen Sie entweder den Befehl update-deployment-group oder create-deployment-group mit den gewünschten Auto Scaling Scaling-Gruppennamen auf. CodeDeployinstalliert die Auto Scaling Scaling-Hooks mit neuen UUIDs neu.

> **Note**
>
> Wenn Sie eine Auto Scaling Scaling-Gruppe von einer CodeDeploy Bereitstellungsgruppe trennen, können alle laufenden Bereitstellungen für die Auto Scaling-Gruppe fehlschlagen, und neue EC2-Instances, die von der Auto Scaling-Gruppe horizontal skaliert werden, erhalten Ihre Anwendungsrevisionen nicht von. CodeDeploy Damit Auto Scaling wieder funktioniert CodeDeploy, müssen Sie die Auto Scaling Scaling-Gruppe erneut der Bereitstellungsgruppe zuordnen und eine neue anrufen, um eine flottenweite Bereitstellung `CreateDeployment` zu starten.

Nicht übereinstimmende Amazon EC2 Auto Scaling Scaling-Lifecycle-Hooks können dazu führen, dass automatische Bereitstellungen für Amazon EC2 Auto Scaling Scaling-Gruppen gestoppt werden oder fehlschlagen

Amazon EC2 Auto Scaling und CodeDeploy verwenden Lifecycle-Hooks, um zu bestimmen, welche Anwendungsrevisionen für welche EC2-Instances bereitgestellt werden sollen, nachdem sie in Amazon EC2 Auto Scaling Scaling-Gruppen gestartet wurden. Automatische Bereitstellungen können gestoppt werden oder fehlschlagen, wenn Lifecycle-Hooks und Informationen zu diesen Hooks in Amazon EC2 Auto Scaling und nicht exakt übereinstimmen. CodeDeploy

Wenn Bereitstellungen für eine Amazon EC2 Auto Scaling-Gruppe fehlschlagen, überprüfen Sie, ob die Lifecycle-Hook-Namen in Amazon EC2 Auto Scaling übereinstimmen. CodeDeploy Wenn nicht, verwenden Sie diese AWS CLI Befehlsaufrufe.

Rufen Sie zunächst die Liste der Lifecycle-Hook-Namen sowohl für die Amazon EC2 Auto Scaling Scaling-Gruppe als auch für die Bereitstellungsgruppe ab:

1. Rufen Sie den Befehl describe-lifelycle-hooks auf und geben Sie den Namen der Amazon EC2 Auto Scaling Scaling-Gruppe an, die der Bereitstellungsgruppe in zugeordnet ist. CodeDeploy Notieren Sie sich jeden `LifecycleHooks`-Wert der Liste `LifecycleHookName` der Ausgabe.
2. Rufen Sie den Befehl get-deployment-group auf und geben Sie den Namen der Bereitstellungsgruppe an, die der Amazon EC2 Auto Scaling Scaling-Gruppe zugeordnet ist.

Suchen Sie in der Ausgabe in der `autoScalingGroups` Liste nach jedem Element, dessen Namenswert mit dem Gruppennamen von Amazon EC2 Auto Scaling übereinstimmt, und notieren Sie sich dann den entsprechenden `hook` Wert.

Vergleichen Sie nun die beiden Mengen von Lebenszyklus-Hook-Namen. Wenn sie Zeichen für Zeichen exakt übereinstimmen, liegt das Problem woanders. Möglicherweise möchten Sie andere Schritte zur Problembehandlung mit Amazon EC2 Auto Scaling ausprobieren, die an anderer Stelle in diesem Abschnitt beschrieben werden.

Wenn jedoch die beiden Mengen von Lebenszyklus-Hook-Namen nicht Zeichen für Zeichen exakt übereinstimmen, führen Sie die folgenden Schritte aus:

1. Wenn es Lebenszyklus-Hook-Namen in der Ausgabe des Befehls describe-lifecycle-hooks gibt, die nicht auch in der Ausgabe des Befehls get-deployment-group vorkommen, führen Sie die folgenden Schritte aus:

   a. Rufen Sie für jeden Lebenszyklus-Hook-Namen in der Ausgabe des Befehls describe-lifecycle-hooks den Befehl delete-lifecycle-hook auf.

   b. Rufen Sie den Befehl update-deployment-group auf und geben Sie den Namen der ursprünglichen Amazon EC2 Auto Scaling Scaling-Gruppe an. CodeDeploy erstellt neue Ersatz-Lifecycle-Hooks in der Amazon EC2 Auto Scaling Scaling-Gruppe und ordnet die Lifecycle-Hooks der Bereitstellungsgruppe zu. Automatische Bereitstellungen sollten jetzt wieder aufgenommen werden, sobald neue Instances zur Amazon EC2 Auto Scaling Scaling-Gruppe hinzugefügt werden.

2. Wenn es Lebenszyklus-Hook-Namen in der Ausgabe des Befehls get-deployment-group gibt, die nicht auch in der Ausgabe des Befehls describe-lifecycle-hooks vorkommen, führen Sie die folgenden Schritte aus:

   a. Rufen Sie den Befehl update-deployment-group auf, geben Sie jedoch nicht den Namen der ursprünglichen Amazon EC2 Auto Scaling Scaling-Gruppe an.

   b. Rufen Sie den update-deployment-group Befehl erneut auf, geben Sie diesmal jedoch den Namen der ursprünglichen Amazon EC2 Auto Scaling Scaling-Gruppe an. CodeDeploy erstellt die fehlenden Lifecycle-Hooks in der Amazon EC2 Auto Scaling Scaling-Gruppe neu. Automatische Bereitstellungen sollten jetzt wieder aufgenommen werden, sobald neue Instances zur Amazon EC2 Auto Scaling Scaling-Gruppe hinzugefügt werden.

Nachdem Sie sichergestellt haben, dass die beiden Gruppen von Lifecycle-Hook-Namen Zeichen für Zeichen exakt übereinstimmen, sollten Anwendungsrevisionen erneut bereitgestellt werden, jedoch nur für neue Instances, die der Amazon EC2 Auto Scaling Scaling-Gruppe hinzugefügt werden. Bereitstellungen erfolgen nicht automatisch für Instances, die sich bereits in der Amazon EC2 Auto Scaling Scaling-Gruppe befinden.

Fehler „Die Bereitstellung ist fehlgeschlagen, weil keine Instances für Ihre Bereitstellungsgruppe gefunden wurden“

Lesen Sie diesen Abschnitt, wenn Sie den folgenden CodeDeploy Fehler sehen:

```
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.
```

Mögliche Ursachen für diesen Fehler sind:

1. Ihre Bereitstellungsgruppeneinstellungen enthalten Tags für EC2-Instances, lokale Instances oder Auto Scaling Scaling-Gruppen, die nicht korrekt sind. Um dieses Problem zu beheben, überprüfen Sie, ob Ihre Tags korrekt sind, und stellen Sie dann Ihre Anwendung erneut bereit.
2. Ihre Flotte wurde nach Beginn der Bereitstellung skaliert. In diesem Szenario werden fehlerfreie Instances in dem `InService` Status Ihrer Flotte angezeigt, es wird jedoch auch der obige Fehler angezeigt. Um dieses Problem zu beheben, stellen Sie Ihre Anwendung erneut bereit.
3. Ihre Auto Scaling Scaling-Gruppe umfasst keine Instances, die sich im `InService` Bundesstaat befinden. Wenn Sie in diesem Szenario versuchen, eine flottenweite Bereitstellung durchzuführen, schlägt die Bereitstellung mit der obigen Fehlermeldung fehl, da sich mindestens eine Instanz im `InService` Status befinden CodeDeploy muss. Es gibt viele Gründe, warum Sie möglicherweise keine Instanzen im `InService` Bundesstaat haben. Einige davon beinhalten:
   - Sie haben die Auto Scaling Scaling-Gruppengröße so geplant (oder manuell konfiguriert)`0`.
   - Auto Scaling hat fehlerhafte EC2-Instances erkannt (z. B. hatten die EC2-Instances Hardwarefehler) und hat sie alle storniert, sodass keine Instances mehr im `InService` Status verbleiben.
   - Während eines Scale-Out-Ereignisses von `0` bis `1` CodeDeploy wurde eine zuvor erfolgreiche Revision (die so genannte letzte erfolgreiche Revision) bereitgestellt, die seit der letzten Bereitstellung nicht mehr funktionsfähig war. Dies führte dazu, dass die Bereitstellung auf der

Scaled-Out-Instance fehlschlug, was wiederum dazu führte, dass Auto Scaling die Instance stornierte und keine Instances im `InService` Status zurückblieben.

Wenn Sie feststellen, dass Sie keine Instances im `InService` Status haben, beheben Sie das Problem wie im folgenden Verfahren beschrieben. To troubleshoot the error if there are no instances in the InService state

Um den Fehler zu beheben, wenn es im InService Bundesstaat keine Instanzen gibt

1. Überprüfen Sie in der Amazon EC2 EC2-Konsole die Einstellung Gewünschte Kapazität. Wenn sie Null ist, setzen Sie sie auf eine positive Zahl. Warten Sie, bis die Instanz fertig ist`InService`, was bedeutet, dass die Bereitstellung erfolgreich war. Sie haben das Problem behoben und können die verbleibenden Schritte dieses Fehlerbehebungsverfahrens überspringen. Informationen zur Einstellung der gewünschten Kapazität finden Sie unter Festlegen von Kapazitätsgrenzen für Ihre Auto Scaling Scaling-Gruppe im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.
2. Wenn Auto Scaling weiterhin versucht, neue EC2-Instances zu starten, um die gewünschte Kapazität zu erreichen, das Scale-Out aber nie erreichen kann, liegt das in der Regel an einem fehlgeschlagenen Auto Scaling Scaling-Lifecycle-Hook. Beheben Sie dieses Problem wie folgt:

   a. Informationen dazu, welches Auto Scaling Scaling-Lifecycle-Hook-Ereignis fehlschlägt, finden Sie unter Verifizieren einer Skalierungsaktivität für eine Auto Scaling Scaling-Gruppe im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

   b. Wenn der Name des fehlgeschlagenen Hooks lautet`CodeDeploy-managed-automatic-launch-deployment-hook-`*`DEPLOYMENT_GROUP_NAME`*, gehen Sie zu CodeDeploy, suchen Sie die Bereitstellungsgruppe und suchen Sie die fehlgeschlagene Bereitstellung, die von Auto Scaling gestartet wurde. Untersuchen Sie dann, warum die Bereitstellung fehlgeschlagen ist.

   c. Wenn Sie wissen, warum die Bereitstellung fehlgeschlagen ist (z. B. weil CloudWatch Alarme aufgetreten sind), und Sie das Problem beheben können, ohne die Version zu ändern, sollten Sie dies jetzt tun.

   d. Wenn Sie nach einer Untersuchung feststellen, dass die CodeDeploy letzte erfolgreiche Revision nicht mehr fehlerfrei ist und es keine fehlerfreien Instances in Ihrer Auto Scaling Scaling-Gruppe gibt, befinden Sie sich in einem Bereitstellungs-Deadlock-Szenario. Um dieses Problem zu lösen, müssen Sie die fehlerhafte CodeDeploy Revision reparieren, indem Sie den Lifecycle-Hook vorübergehend aus der Auto Scaling Scaling-Gruppe

entfernen CodeDeploy, den Hook dann erneut installieren und eine neue (gute) Revision erneut bereitstellen. Detaillierte Anweisungen finden Sie unter:

- To fix the deployment deadlock issue (CLI)
- To fix the deployment deadlock issue (console)

So beheben Sie das Deployment-Deadlock-Problem (CLI)

1. (Optional) Blockieren Sie Ihre CI/CD Pipelines, die den CodeDeploy Fehler verursachen, sodass es nicht zu unerwarteten Bereitstellungen kommt, während Sie dieses Problem beheben.
2. Notieren Sie sich Ihre aktuelle Auto Scaling DesiredCapacityScaling-Einstellung:

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name ASG_NAME
   ```

   Möglicherweise müssen Sie am Ende dieses Vorgangs wieder auf diese Zahl zurückskalieren.
3. Stellen Sie Ihre Auto Scaling DesiredCapacityScaling-Einstellung auf ein1. Dies ist optional, wenn Ihre gewünschte Kapazität höher war als 1 ursprünglich angegeben. Wenn Sie sie auf reduzieren1, nimmt die Instanz weniger Zeit für die Bereitstellung und spätere Bereitstellung in Anspruch, was die Fehlerbehebung beschleunigt. Wenn Ihre gewünschte Auto Scaling-Kapazität ursprünglich auf eingestellt war0, müssen Sie sie auf erhöhen1. Dies ist verpflichtend.

   aws autoscaling set-desired-capacity --auto-scaling-group-name --desired-capacity 1 *ASG_NAME*

   > **Note**
   >
   > Bei den verbleibenden Schritten dieses Verfahrens wird davon ausgegangen, dass Sie Ihren Wert auf eingestellt haben. DesiredCapacity1

   An diesem Punkt versucht Auto Scaling, auf eine Instanz zu skalieren. Da der CodeDeploy hinzugefügte Hook immer noch vorhanden ist, CodeDeploy versucht er dann die Bereitstellung; die Bereitstellung schlägt fehl; Auto Scaling bricht die Instance ab; und Auto Scaling versucht, eine Instance neu zu starten, um die gewünschte Kapazität von einer Instanz zu erreichen, was wiederum fehlschlägt. Sie befinden sich in einer Abbruch-Relaunch-Schleife.
4. De-register die Auto Scaling Scaling-Gruppe aus der Bereitstellungsgruppe:

> **Warning**
>
> Mit dem folgenden Befehl wird eine neue EC2-Instance ohne Software gestartet. Bevor Sie den Befehl ausführen, stellen Sie sicher, dass eine Auto Scaling `InService` Scaling-Instanz, auf der keine Software ausgeführt wird, akzeptabel ist. Stellen Sie beispielsweise sicher, dass der mit der Instance verknüpfte Load Balancer ohne Software keinen Datenverkehr an diesen Host sendet.

> **Important**
>
> Verwenden Sie den unten abgebildeten CodeDeploy Befehl, um den Hook zu entfernen. Entfernen Sie den Hook nicht über den Auto Scaling Scaling-Dienst, da das Entfernen von nicht erkannt wird CodeDeploy.

```
aws deploy update-deployment-group --application-name APPLICATION_NAME
--current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-
groups
```

Nach der Ausführung dieses Befehls passiert Folgendes:

a. CodeDeploy Deregistriert die Auto Scaling Scaling-Gruppe von der Bereitstellungsgruppe ab.

b. CodeDeploy entfernt den Auto Scaling Scaling-Lifecycle-Hook aus der Auto Scaling Scaling-Gruppe.

c. Da der Hook, der eine fehlgeschlagene Bereitstellung verursacht hat, nicht mehr vorhanden ist, storniert Auto Scaling die bestehende EC2-Instance und startet sofort eine neue, um auf die gewünschte Kapazität zu skalieren. Die neue Instance sollte bald ihren Status annehmen. `InService` Die neue Instanz enthält keine Software.

5. Warten Sie, bis die EC2-Instance den `InService` Status erreicht hat. Verwenden Sie den folgenden Befehl, um ihren Status zu überprüfen:

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
   ASG_NAME --query AutoScalingGroups[0].Instances[*].LifecycleState
   ```

6. Fügen Sie den Hook zurück zur EC2-Instance hinzu:

> **Important**
>
> Verwenden Sie den unten abgebildeten CodeDeploy Befehl, um den Hook hinzuzufügen. Verwenden Sie nicht den Auto Scaling Scaling-Dienst, um den Hook hinzuzufügen, da der Zusatz von nicht erkannt wird CodeDeploy.

```
aws deploy update-deployment-group --application-name APPLICATION_NAME
--current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-
groups ASG_NAME
```

Nach der Ausführung dieses Befehls passiert Folgendes:

a. CodeDeploy installiert den Auto Scaling Scaling-Lifecycle-Hook erneut für die EC2-Instance

b. CodeDeploy registriert die Auto Scaling Scaling-Gruppe erneut bei der Bereitstellungsgruppe.

7. Erstellen Sie eine flottenweite Bereitstellung mit Amazon S3 oder einer GitHub Version, von der Sie wissen, dass sie fehlerfrei ist und die Sie verwenden möchten.

   Wenn es sich bei der Revision beispielsweise um eine ZIP-Datei in einem Amazon S3 S3-Bucket handelt, der `my-revision-bucket` mit dem Objektschlüssel von aufgerufen wird`httpd_app.zip`, geben Sie den folgenden Befehl ein:

   ```
   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}"
   ```

   Da es jetzt eine `InService` Instanz in der Auto Scaling Scaling-Gruppe gibt, sollte diese Bereitstellung funktionieren, und der Fehler Die Bereitstellung ist fehlgeschlagen, weil für Ihre Bereitstellungsgruppe keine Instanzen gefunden wurden, sollte nicht mehr angezeigt werden.

8. Wenn die Bereitstellung erfolgreich war, skalieren Sie Ihre Auto Scaling Scaling-Gruppe wieder auf die ursprüngliche Kapazität, falls Sie sie zuvor skaliert haben:

   ```
   aws autoscaling set-desired-capacity --auto-scaling-group-name ASG_NAME
   --desired-capacity ORIGINAL_CAPACITY
   ```

So beheben Sie das Deadlock-Problem bei der Bereitstellung (Konsole)

1. (Optional) Sperren Sie Ihre CI/CD Pipelines, die den CodeDeploy Fehler verursachen, sodass es nicht zu unerwarteten Bereitstellungen kommt, während Sie dieses Problem beheben.
2. Gehen Sie zur Amazon EC2 EC2-Konsole und notieren Sie sich Ihre gewünschte Kapazitätseinstellung für Auto Scaling. Möglicherweise müssen Sie am Ende dieses Verfahrens wieder auf diese Zahl zurückskalieren. Informationen zum Auffinden dieser Einstellung finden Sie unter Kapazitätsgrenzen für Ihre Auto Scaling Scaling-Gruppe festlegen.
3. Stellen Sie die gewünschte Anzahl von EC2-Instances wie folgt ein1:

   Dies ist optional, wenn Ihre gewünschte Kapazität höher war als 1 ursprünglich angegeben. Wenn Sie sie auf reduzieren1, nimmt die Instanz weniger Zeit für die Bereitstellung und spätere Bereitstellung in Anspruch, was die Fehlerbehebung beschleunigt. Wenn Ihre gewünschte Auto Scaling-Kapazität ursprünglich auf eingestellt war`0`, müssen Sie sie auf erhöhen`1`. Dies ist verpflichtend.

   > **Note**
   >
   > Bei den verbleibenden Schritten dieses Verfahrens wird davon ausgegangen, dass Sie Ihre gewünschte Kapazität auf eingestellt haben`1`.

   a. Öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/und wählen Sie im Navigationsbereich Auto Scaling Groups aus.

   b. Wählen Sie die entsprechende Region aus.

   c. Gehen Sie zur problematischen Auto Scaling Scaling-Gruppe.

   d. Wählen Sie in den Gruppendetails die Option Bearbeiten aus.

   e. Stellen Sie Gewünschte Kapazität auf ein**1**.

   f. Wählen Sie Aktualisieren aus.
4. De-register die Auto Scaling Scaling-Gruppe aus der Bereitstellungsgruppe:

   > **Warning**
   >
   > Mit den folgenden Teilschritten wird eine neue EC2-Instance ohne Software gestartet. Bevor Sie den Befehl ausführen, stellen Sie sicher, dass eine Auto Scaling `InService` Scaling-Instanz, auf der keine Software ausgeführt wird, akzeptabel ist. Stellen Sie

beispielsweise sicher, dass der mit der Instance verknüpfte Load Balancer ohne Software keinen Datenverkehr an diesen Host sendet.

a. Öffnen Sie die CodeDeploy Konsole unter. https://console.aws.amazon.com/codedeploy/
b. Wählen Sie die entsprechende Region aus.
c. Wählen Sie im Navigationsbereich Applications (Anwendungen).
d. Wählen Sie den Namen Ihrer CodeDeploy Anwendung.
e. Wählen Sie den Namen Ihrer CodeDeploy Bereitstellungsgruppe.
f. Wählen Sie Bearbeiten aus.
g. Deaktivieren Sie in der Umgebungskonfiguration die Amazon EC2 Auto Scaling Scaling-Gruppen.

> **Note**
>
> In der Konsole können Sie die Konfiguration nicht speichern, wenn keine Umgebungskonfiguration definiert ist. Um die Prüfung zu umgehen, fügen Sie vorübergehend ein Tag von `EC2` oder hinzu, von dem Sie wissen`On-premises`, dass es nicht zu Hosts aufgelöst wird. Um ein Tag hinzuzufügen, wählen Sie Amazon EC2 EC2-Instances oder On-premises -Instance aus und fügen Sie einen Tag-Schlüssel von **`EC2`** oder **`On-premises`** hinzu. Sie können das Tag Value leer lassen.

h. Wählen Sie Änderungen speichern aus.

Nach Abschluss dieser Teilschritte geschieht Folgendes:

i. CodeDeploy Deregistriert die Auto Scaling Scaling-Gruppe von der Bereitstellungsgruppe ab.
ii. CodeDeploy entfernt den Auto Scaling Scaling-Lifecycle-Hook aus der Auto Scaling Scaling-Gruppe.
iii. Da der Hook, der eine fehlgeschlagene Bereitstellung verursacht hat, nicht mehr vorhanden ist, storniert Auto Scaling die bestehende EC2-Instance und startet sofort eine neue, um auf die gewünschte Kapazität zu skalieren. Die neue Instance sollte bald ihren Status annehmen. `InService` Die neue Instanz enthält keine Software.

5. Warten Sie, bis die EC2-Instance den `InService` Status erreicht hat. Um ihren Status zu überprüfen:

   a. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/.

   b. Wählen Sie im Navigationsbereich die Option Auto Scaling Groups (Gruppen).

   c. Wählen Sie Ihre Auto Scaling Scaling-Gruppe.

   d. Wählen Sie im Inhaltsbereich die Registerkarte Instance Management aus.

   e. Vergewissern Sie sich, dass unter Instances in der Spalte Lifecycle InServiceneben der Instanz angezeigt wird.

6. Re-register die Auto Scaling Scaling-Gruppe mit der CodeDeploy Bereitstellungsgruppe, die dieselbe Methode verwendet, mit der Sie sie entfernt haben:

   a. Öffnen Sie die CodeDeploy Konsole unter https://console.aws.amazon.com/codedeploy/.

   b. Wählen Sie die entsprechende Region aus.

   c. Wählen Sie im Navigationsbereich Applications (Anwendungen).

   d. Wählen Sie den Namen Ihrer CodeDeploy Anwendung.

   e. Wählen Sie den Namen Ihrer CodeDeploy Bereitstellungsgruppe.

   f. Wählen Sie Bearbeiten aus.

   g. Wählen Sie in der Umgebungskonfiguration Amazon EC2 Auto Scaling Scaling-Gruppen und wählen Sie Ihre Auto Scaling-Gruppe aus der Liste aus.

   h. Suchen Sie unter Amazon EC2 EC2-Instances oder On-premises-Instances nach dem Tag, den Sie hinzugefügt haben, und entfernen Sie es.

   i. Deaktivieren Sie das Kontrollkästchen neben Amazon EC2 EC2-Instances oder On-premises Instances.

   j. Wählen Sie Änderungen speichern aus.

   Diese Konfiguration installiert den Lifecycle-Hook erneut in der Auto Scaling Scaling-Gruppe.

7. Erstellen Sie eine flottenweite Bereitstellung mit Amazon S3 oder einer GitHub Version, von der Sie wissen, dass sie fehlerfrei ist und die Sie verwenden möchten.

   Wenn es sich bei der Revision beispielsweise um eine ZIP-Datei in einem Amazon S3 S3-Bucket handelt, der `my-revision-bucket` mit dem Objektschlüssel von aufgerufen wird`httpd_app.zip`, gehen Sie wie folgt vor:

a. Wählen Sie in der CodeDeploy Konsole auf der Seite Deployment Group die Option Create deployment aus.
b. Wählen Sie unter Revision type (Revisionstyp) die Option My application is stored in Amazon S3 (Meine Anwendung ist in Amazon S3 gespeichert) aus.
c. Wählen Sie als Speicherort der Version die Option `s3://my-revision-bucket/httpd_app.zip`.
d. Wählen Sie als Revisionsdateityp die Option `.zip`.
e. Wählen Sie Create deployment.

Da es jetzt eine `InService` Instanz in der Auto Scaling Scaling-Gruppe gibt, sollte diese Bereitstellung funktionieren, und der Fehler Die Bereitstellung ist fehlgeschlagen, weil für Ihre Bereitstellungsgruppe keine Instanzen gefunden wurden, sollte nicht mehr angezeigt werden.

8. Wenn die Bereitstellung erfolgreich war, skalieren Sie Ihre Auto Scaling Scaling-Gruppe wieder auf die ursprüngliche Kapazität, falls Sie sie zuvor skaliert haben:

a. Öffnen Sie die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/und wählen Sie im Navigationsbereich Auto Scaling Groups aus.
b. Wählen Sie die entsprechende Region aus.
c. Gehen Sie zu Ihrer Auto Scaling Scaling-Gruppe.
d. Wählen Sie in den Gruppendetails die Option Bearbeiten aus.
e. Setzen Sie die gewünschte Kapazität auf ihren ursprünglichen Wert zurück.
f. Wählen Sie Aktualisieren.

Fehlercodes für AWS CodeDeploy

Dieses Thema enthält Referenzinformationen zu CodeDeploy Fehlern.

Fehlercode	Description
`AGENT_ISSUE`	Die Bereitstellung ist aufgrund eines Problems mit dem CodeDeploy Agenten fehlgeschlagen. Stellen Sie sicher, dass der Agent in allen Instances in der Bereitstellungsgruppe installiert ist und ausgeführt wird.

Fehlercode	Description
	Weitere Informationen: • Stellen Sie sicher, dass der CodeDeploy Agent läuft • Installieren Sie den CodeDeploy Agenten • Mit dem CodeDeploy Agenten arbeiten
`AUTO_SCALING_IAM_ROLE_PERMISSIONS`	Die Ihrer Bereitstellungsgruppe zugeordnete Servicerolle verfügt nicht über die erforderlichen Berechtigungen, um Vorgänge im folgenden AWS Dienst auszuführen. Weitere Informationen: • Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy • Eine Rolle zum Delegieren von Berechtigungen an einen AWS Dienst erstellen
`HEALTH_CONSTRAINTS`	Die Gesamtbereitstellung ist fehlgeschlagen, weil die Revision in zu vielen Einzel-Instances nicht bereitgestellt werden konnte, weil zu wenige funktionsfähige Instances für eine Bereitstellung verfügbar sind oder weil bei einigen Instances in der Bereitstellungsgruppe Probleme aufgetreten sind. Weitere Informationen: • Instance Health • Beheben von Instance-Problemen • Beheben Sie Probleme bei der EC2/On-Premises Bereitstellung

Fehlercode	Description
HEALTH_CONSTRAINTS_INVALID	Die Bereitstellung kann nicht gestartet werden, da die Mindestanzahl fehlerfreier Instanzen, wie in Ihrer Bereitstellungskonfiguration definiert , nicht verfügbar ist. Sie können die erforderl iche Anzahl an funktionsfähigen Instances durch eine Aktualisierung der Bereitste llungskonfiguration reduzieren oder die Anzahl der Instances in der Bereitstellungsgruppe erhöhen. Weitere Informationen: • Instance Health • Arbeiten mit Instanzen für CodeDeploy
IAM_ROLE_MISSING	Die Bereitstellung ist fehlgeschlagen, da keine Service-Rolle mit dem Namen der für die Bereitstellungsgruppe angegebenen Service-Rolle existiert. Vergewissern Sie sich, dass Sie den Namen der Service-Rolle korrekt angegeben haben. Weitere Informationen: • Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy • Ändern Sie die Einstellungen der Bereitste llungsgruppe mit CodeDeploy

Fehlercode	Description
`IAM_ROLE_PERMISSIONS`	CodeDeploy verfügt nicht über die erforderlichen Berechtigungen, um eine Rolle zu übernehmen, oder die von Ihnen verwendete IAM-Rolle gibt Ihnen nicht die Erlaubnis, Operationen in einem AWS Dienst auszuführen. Weitere Informationen: • Schritt 1: Einrichtung • Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy • Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances

Fehlercode	Description
NO_INSTANCES	Dies kann durch eine der folgenden Ursachen bedingt sein. • Wenn Sie Amazon EC2-Tags für eine EC2/On-Premises blue/green Bereitstellung verwenden, sind diese möglicherweise nicht richtig konfiguriert. Stellen Sie in Ihrer CodeDeploy Bereitstellungsgruppe sicher, dass sie in Ihren blauen Instances und Ihren grünen Instances enthalten sind. Sie können mithilfe der Amazon EC2 EC2-Konsole überprüfen, ob Ihre Instances ordnungsg emäß gekennzeichnet sind. • Wenn Sie eine Amazon EC2 Auto Scaling Scaling-Gruppe verwenden, verfügt Ihre Auto Scaling-Gruppe möglicherweise nicht über genügend Kapazität. Stellen Sie sicher, dass Ihre Auto Scaling Scaling-Gruppe über genügend Kapazität für Ihre Bereitste llung verfügt. Sie können die Kapazität Ihrer Amazon EC2 Auto Scaling Scaling-Gruppe anhand der Anzahl fehlerfreier Instances in der Amazon EC2-Konsole einsehen. • Bei einer EC2/On-Premises blue/green Bereitstellung haben die blaue und die grüne Flotte möglicherweise nicht dieselbe Größe. Stellen Sie sicher, dass beide Flotten dieselbe Größe haben. Weitere Informationen: • Tagging Instances for Deployments

Fehlercode	Description
	• Tutorial: Wird verwendet CodeDeploy , um eine Anwendung für eine Auto Scaling Scaling-Gruppe bereitzustellen • Erstellen Sie eine Anwendung für eine blue/green Bereitstellung (Konsole)
OVER_MAX_INSTANCES	Die Bereitstellung ist fehlgeschlagen, weil Sie die Revision in mehr Instances bereitstellen möchten, als für Ihr Konto zulässig ist. Aktualisieren Sie die Tag-Einstellungen für die Bereitstellungsgruppe oder löschen Sie einige der Ziel-Instances, um die Anzahl der Ziel-Instances für die Bereitstellung zu reduzieren. Sie können sich auch an uns wenden, AWS Support um eine Erhöhung des Limits zu beantragen. Weitere Informationen: • Ändern Sie die Einstellungen der Bereitstellungsgruppe mit CodeDeploy • CodeDeploy Kontingente • Limit-Erhöhung anfordern
THROTTLED	Die Bereitstellung schlug fehl, weil mehr Anfragen gestellt wurden, als AWS CodeDeploy von einer IAM-Rolle zugelassen wurden. Reduzieren Sie die Anzahl der Anforderungen. Weitere Informationen: • Anfragerate der Anfrage-API

Fehlercode	Description
UNABLE_TO_SEND_ASG	Die Bereitstellung ist fehlgeschlagen, weil die Bereitstellungsgruppe nicht korrekt mit ihrer Amazon EC2 Auto Scaling Scaling-Gruppe konfiguriert ist. Löschen Sie in der CodeDeploy Konsole die Amazon EC2 Auto Scaling Scaling-Gruppe aus der Bereitstellungsgruppe und fügen Sie sie dann erneut hinzu. Weitere Informationen: • Unter der Haube: CodeDeploy und Auto Scaling Scaling-Integration

Verwandte Themen

Problembehebung CodeDeploy

CodeDeploy Ressourcen

Die folgenden verwandten Ressourcen können Ihnen bei der Arbeit mit helfen CodeDeploy.

Referenzhandbücher und Support-Ressourcen

- AWS CodeDeploy API-Referenz — Beschreibungen, Syntax und Anwendungsbeispiele zu CodeDeploy Aktionen und Datentypen, einschließlich allgemeiner Parameter und Fehlercodes.
- CodeDeploy Häufig gestellte technische Fragen — Die häufigsten Fragen von Kunden zu CodeDeploy.
- AWS Support Center — Die zentrale Anlaufstelle für die Erstellung und Verwaltung Ihrer AWS Support Fälle. Enthält auch Links zu anderen Ressourcen wie Foren, häufig gestellten technischen Fragen, dem Status des Dienstes und AWS Trusted Advisor.
- AWS Supportpläne — Die wichtigste Webseite mit Informationen zu AWS Support Plänen.
- Kontaktiere uns — Eine zentrale Anlaufstelle für Anfragen zu AWS Rechnungen, Konten, Veranstaltungen, Missbrauch und anderen Problemen.
- AWS Nutzungsbedingungen der Website — Detaillierte Informationen zu unseren Urheberrechten und Marken, zu Ihrem Konto, Ihrer Lizenz und Ihrem Zugriff auf die Website sowie zu anderen Themen.

Beispiele

- CodeDeploy Beispiele auf GitHub — Beispiele und Vorlagenszenarien für CodeDeploy.
- CodeDeploy Jenkins-Plugin — Jenkins-Plugin für. CodeDeploy
- CodeDeploy Agent — Open-source Version des Agenten. CodeDeploy

Blogs

- AWS DevOps Blog — Einblicke für Entwickler, Systemadministratoren und Architekten.

AWS Kits und Tools für die Softwareentwicklung

Die folgenden AWS SDKs und Tools unterstützen die Lösungsentwicklung mit CodeDeploy:

- AWS SDK für .NET
- AWS SDK für Java
- AWS SDK für JavaScript
- AWS SDK für PHP
- AWS SDK für Python (Boto)
- AWS SDK für Ruby
- AWS Toolkit for Eclipse — Teile 1, 2 und 3.
- AWS Tools for Windows PowerShell— Eine Reihe von PowerShell Windows-Cmdlets, die die Funktionalität von AWS SDK für .NET in der PowerShell Umgebung verfügbar machen.
- CodeDeploy Cmdlets in AWS -Tools für PowerShell — Eine Reihe von PowerShell Windows-Cmdlets, die die Funktionalität von in der Umgebung verfügbar machen. CodeDeploy PowerShell
- AWS Command Line Interface— Eine einheitliche Befehlszeilensyntax für den Zugriff auf Dienste. AWS Der AWS CLI verwendet einen einzigen Einrichtungsprozess, um den Zugriff für alle unterstützten Dienste zu ermöglichen.
- AWS Entwicklertools — Links zu Entwicklertools und Ressourcen mit Dokumentation, Codebeispielen, Versionshinweisen und anderen Informationen, die Sie bei der Entwicklung innovativer Anwendungen mit CodeDeploy und unterstützen AWS.

Dokumentverlauf

Die folgende Tabelle beschreibt die wichtigsten Änderungen in diesem Benutzerhandbuch zur Unterstützung der neuen und verbesserten Funktionen seit der letzten Version des CodeDeploy - Benutzerhandbuchs.

- API-Version: 2014-10-06

Änderung	Beschreibung	Datum
CodeDeploy Agent-Version v2.1.2 mit macOS-Unterstützung	Der AWS CodeDeploy Agent wurde auf Version 2.1.2 aktualisiert, wodurch Unterstützung für macOS auf EC2-Mac-Instances hinzugefügt wird. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten und Installieren des CodeDeploy Agenten für macOS.	13. Oktober 2026
CodeDeploy Agent-Version v2.1.1	Der AWS CodeDeploy Agent wurde auf Version 2.1.1 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	2. Oktober 2026
CodeDeploy Agent-Version v2.1.0	Der AWS CodeDeploy Agent wurde auf Version 2.1.0 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	7. September 2026
CodeDeploy Agent-Version v2.0.1	Der AWS CodeDeploy Agent wurde auf Version	24. August 2026

	2.0.1 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	
CodeDeploy Agent-Version v2.0.0	Der AWS CodeDeploy Agent wurde auf Version 2.0.0 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	15. Juli 2026
CodeDeploy Version v1.8.1 für Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.8.1 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	3. Februar 2026
CodeDeploy Version v1.8.0 für Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.8.0 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	31. Juli 2025
CodeDeploy hat eine bestehende AWS verwaltete Richtlinie aktualisiert	Das `AWSCodeDeployDeployerAccess` wurde aktualisiert. Weitere Informationen finden Sie unter AWS Verwaltete Richtlinienaktualisierungen.	16. Dezember 2024

CodeDeploy hat eine bestehende AWS verwaltete Richtlinie aktualisiert	Das `AWSCodeDeployReadOnlyAccess` wurde aktualisiert. Weitere Informationen finden Sie unter AWS Verwaltete Richtlinienaktualisierungen.	16. Dezember 2024
CodeDeploy hat eine bestehende AWS verwaltete Richtlinie aktualisiert	Das `AWSCodeDeployFullAccess` wurde aktualisiert. Weitere Informationen finden Sie unter AWS Verwaltete Richtlinienaktualisierungen.	16. Dezember 2024
CodeDeploy Version v1.7.1 für Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.7.1 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	14. November 2024
Aktualisierte Amazon S3 S3-Bucket-Namen	Das Beispiel für Amazon S3 S3-Buckets in diesem Handbuch wurde aktualisiert, sodass nun Namen verwendet werden, die von AWS reserviert sind.	17. Juni 2024
Alternativer Text (Alt-Text) hinzugefügt	Alle Bilder in diesem Handbuch wurden aktualisiert und enthalten nun Alternativtext. Alt-Text wird von Screenreadern gelesen und macht unsere Dokumentation für blinde Benutzer zugänglicher.	22. Mai 2024

CodeDeploy Version v1.7.0 für Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.7.0 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	6. März 2024
Befehle wurden geändert	Die `sudo service codedeploy-agent status|start|stop` Befehle werden nicht mehr empfohlen, da sie nicht `systemd` für das Prozessmanagement von CodeDeploy Agenten verwendet werden, was eine bewährte Methode ist. Um sicherzustellen, `systemd` dass sie verwendet werden, verwenden Sie `systemctl` Befehle, wie im folgenden Beispiel gezeigt:`sytemctl start codedeploy-agent` . Die folgenden Themen wurden mit `systemctl` Befehlen aktualisiert: Den CodeDeploy Agenten für Amazon Linux oder RHEL installieren, Den CodeDeploy Agenten für Ubuntu Server installieren, Fehler bei allen übersprungenen Lebenszyklusereignissen beheben und Eine neue CodeDeploy Protokolldatei erstellen, falls sie versehentlich gelöscht wurde.	12. Januar 2024

Themen wurden hinzugefügt	Die Themen Verwaltung des CodeDeploy Agentenprozesses und Referenzierung von Dateien in Ihren Lebenszyklus-Ereignisskripten wurden hinzugefügt.	12. Januar 2024
CodeDeploy unterstützt jetzt zonale Konfigurationen	Das CodeDeploy Thema Bereitstellungskonfiguration mit Informationen zur zonalen Konfiguration erstellen wurde aktualisiert.	07. Dezember 2023
CodeDeploy unterstützt jetzt Bereitstellungen zur Terminierung	Das Thema Aktivierung von Terminierungsbereitstellungen bei Auto Scaling-Scale-In-Ereignissen wurde hinzugefügt, um die Funktion zur Terminierung der Bereitstellung zu beschreiben. Außerdem wurden die Themen AppSpec „Hooks“ für eine EC2/On-Premises Bereitstellung, Erstellen einer Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole) und Erstellen einer Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung (Konsole) aktualisiert, um diese Funktion zu berücksichtigen.	07. Dezember 2023

JSON-Formatierung behoben	Die Formatierung der JSON-Codebeispiele im Abschnitt AppSpec „Ressourcen“ (nur Amazon ECS und AWS Lambda Bereitstellungen) wurde behoben.	3. Dezember 2023
Ein Thema zur Fehlerbehebung wurde hinzugefügt	Es wurde ein Thema zur Behebung von Problemen bei der Bereitstellung von Amazon ECS hinzugefügt.	24. Oktober 2023
Der AppSpec Dateiname wurde aktualisiert	Die CodeDeploy AppSpec Dateireferenz wurde aktualisiert und gibt nun an, dass die AppSpec Datei `appspec.yml` für EC2/On-Premises Bereitstellungen benannt werden muss.	05. Oktober 2023
CodeDeploy unterstützt jetzt mehrere Load Balancer	Die Themen Eine Bereitstellungsgruppe für eine direkte Bereitstellung erstellen (Konsole), Eine Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung erstellen (Konsole) und Einen Load Balancer einrichten in Elastic Load Balancing für CodeDeploy Amazon EC2 EC2-Bereitstellungen wurden aktualisiert, um darauf hinzuweisen, dass mehrere Load Balancer unterstützt werden.	26. September 2023

Das Thema Regionen im VPC wurde aktualisiert	Die Tabelle im Thema Use CodeDeploy with Amazon Virtual Private Cloud wurde aktualisiert und zeigt nun zusätzliche Unterstützung für Regionen an. Insbesondere die Regionen Asien-Pazifik (Hyderabad), Asien-Pazifik (Melbourne), Europa (Mailand), Europa (Spanien) und Europa (Zürich) wurden aktualisiert und bieten nun Unterstützung für den Agenten-Endpunkt.	22. September 2023
Die Regionen wurden im Resource-Kit-Thema aktualisiert	Dem Abschnitt Bucket-Namen nach AWS Regionen im Resource Kit wurden die folgenden Regionen hinzugefügt: Asien-Pazifik (Osaka), Asien-Pazifik (Hyderabad), Kanada (Zentral), Europa (Spanien), Europa (Zürich), Naher Osten (VAE). Außerdem wurden die IAM-Richtlinien für diese Regionen und alle anderen, die noch fehlten, aktualisiert.	22. September 2023

Verkürzte Themen zur Installation und Aktualisierung von Agenten	Die Themen „Den CodeDeploy Agenten für Windows Server installieren“ und „ CodeDeploy Agent auf Windows Server aktualisieren“ wurden gekürzt. Redundante Amazon S3 S3-Bucket-URLs und Amazon S3 S3-Kopierbefehle wurden entfernt.	21. September 2023
Die Region Asien-Pazifik (Jakarta) wurde hinzugefügt	Asien-Pazifik (Jakarta) wurde zu den Bucket-Namen des Resource Kits nach Region hinzugefügt.	21. September 2023
CodeDeploy hat eine bestehende AWS verwaltete Richtlinie aktualisiert	Die AWSCodeDeployRole verwaltete Richtlinie wurde aktualisiert. Weitere Informationen finden Sie unter AWS - Aktualisierungen von AWS - verwalteten Richtlinien.	16. August 2023
Ein Limit wurde hinzugefügt	Dem Thema CodeDeploy Limits wurde ein Limit hinzugefügt. Das Limit ist Die maximale Anzahl von Alarmen, die einer Bereitstellungsgruppe zugeordnet sind.	15. August 2023
Ein Schritt im Zusammenhang mit Load Balancern wurde behoben	Die Anweisungen unter Erstellen einer Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung (Konsole) wurden behoben. Der Load Balancer-Schritt ist jetzt als optional gekennzeichnet.	3. August 2023

Klarer Wortlaut im Amazon ECS-Thema	Der Wortlaut im Tutorial: Bereitstellen einer Anwendung in Amazon ECS wurde klargestellt. Der Wortlaut weist jetzt darauf hin, dass Sie eine Anwendung bereitstellen. Zuvor deutete der Wortlaut darauf hin, dass Sie einen Amazon ECS-Service einsetzen.	3. August 2023
CodeDeploy ist jetzt in der Region Israel (Tel Aviv) verfügbar	CodeDeploy ist jetzt in der Region Israel (Tel Aviv) (il-central-1) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen.	31. Juli 2023
Aktualisierung des Themas	Das Thema Beheben von EC2/On-Premises Bereitstellungsproblemen wurde um einen Tipp zur automatischen Ausführung von Problembehandlungsaufgaben mithilfe eines Runbooks aktualisiert.	07. Juli 2023
Das Thema wurde aktualisiert	Der Abschnitt AppSpec „Ressourcen“ für das Thema Amazon ECS-Bereitstellungen wurde mit weiteren Informationen zur Aufgabendefinition ARN aktualisiert.	07. Juli 2023

Aktualisierung des Themas	Das Thema Schritt 1: Installation und Konfiguration AWS CLI der lokalen Instanz wurde mit Informationen zur Fehlerbehebung aktualisiert.	07. Juli 2023
Das Thema wurde aktualisiert	Das Thema Cross-serviceConfused Deputy Prevention wurde mit Informationen über Amazon blue/green ECS-Bereitstellungen aktualisiert. CloudFormation	6. Juli 2023
Das Thema wurde aktualisiert	Das Thema Cross-serviceConfused Deputy Prevention wurde durch CloudFormation mit Informationen über Amazon blue/green ECS-Bereitstellungen aktualisiert.	6. Juli 2023
Das Thema wurde aktualisiert	Das Thema Vordefinierte Bereitstellungskonfigurationen für eine EC2/On-Premises Rechenplattform wurde aktualisiert. Es wurde ein Hinweis zum Verhalten der `CodeDeployDefault.HalfAtATime` vordefinierten Bereitstellungskonfiguration mit Auto Scaling Scaling-Gruppen hinzugefügt.	29. Juni 2023

Aktualisierung des Themas	Das AWS CodeDeploy Thema Infrastruktursicherheit wurde aktualisiert, um die neuen Mindestversionen und empfohlenen Versionen des Transport Layer Security (TLS) -Protokolls anzugeben.	28. Juni 2023
Aktualisierung einschränken	Das folgende Limit wurde geändert: „Maximale Anzahl von Stunden, die eine Bereitstellung EC2/On-Premises vor Ort ausgeführt werden kann“. Weitere Informationen finden Sie unter Grenzwerte	27. Juni 2023
Aktualisierung des Themas	Das Thema Schritt 3: CodeDeploy Benutzerberechtigungen einschränken wurde mit detaillierten Anweisungen aktualisiert.	31. Mai 2023
CodeDeploy hat eine bestehende AWS verwaltete Richtlinie aktualisiert	Die AWSCodeDeployFullAccess verwaltete Richtlinie wurde aktualisiert. Weitere Informationen finden Sie unter AWS -Aktualisierungen von AWS -verwalteten Richtlinien.	16. Mai 2023
CodeDeploy Version v1.6.0 für Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.6.0 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	30. März 2023

CodeDeploy Version v1.5.0 für den Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.5.0 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	03. März 2023
Aktualisierung der Amazon ECS-Rechenplattform	Bereitstellungen auf einer Amazon ECS-Rechenplattform werden jetzt in der Region Asien-Pazifik (Jakarta) unterstützt.	8. Februar 2023
CodeDeploy hat eine bestehende verwaltete Richtlinie AWS aktualisiert	Die AWSCodeDeployRole verwaltete Richtlinie wurde aktualisiert. Weitere Informationen finden Sie unter AWS - Aktualisierungen von AWS - verwalteten Richtlinien.	3. Februar 2023
Aktualisierung des Themas	Das Thema Verwendung CodeDeploy mit Amazon Virtual Private Cloud wurde mit neuen und geänderten AWS Regionen aktualisiert.	2. Februar 2023
Aktualisierungen des Themas	CodeDeploy ist jetzt in der Region Asien-Pazifik (Melbourne) (ap-southeast-4) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Regionen Rechnung zu tragen.	26. Januar 2023

Updates zu bewährten Sicherheitsmethoden	Der CodeDeploy Abschnitt Erste Schritte mit und einige andere Abschnitte wurden aktualisiert, um den bewährten AWS Sicherheitsmethoden zu entsprechen.	23. Januar 2023
CodeDeploy Version v1.4.1 für Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.4.1 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	6. Dezember 2022
Ein Thema zur Fehlerbehebung wurde hinzugefügt	Es wurde ein Thema zur Behebung von Fehlern hinzugefügt, die durch lange Dateipfade verursacht werden, die mit dem CodeDeploy Agenten für Windows verwendet werden. Weitere Informationen finden Sie unter Lange Dateipfade verursachen die Fehlermeldung „Datei oder Verzeichnis nicht gefunden“.	6. Dezember 2022
Ein Limit wurde geändert	Das folgende Limit wurde geändert: „Maximale Anzahl von benutzerdefinierten Bereitstellungskonfigurationen, die einem AWS Konto zugeordnet sind“. Das Limit ist jetzt 200. Weitere Informationen zu Grenzwerten finden Sie im Thema Grenzwerte.	7. September 2022

CodeDeploy Version v1.4.0 für Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.4.0 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	31. August 2022
Ein paar Beschränkungen wurden behoben.	Die folgenden Grenzwerte wurden behoben: „Die maximale Anzahl gleichzeitiger Bereitstellungen, die einem AWS Konto zugeordnet sind“, beträgt jetzt 1000. Die 'Maximale Anzahl von Instanzen in einer einzelnen Bereitstellung' ist jetzt 1000. 'Die maximale Anzahl von Instanzen, die von gleichzeitigen Bereitstellungen verwendet werden können, die gerade ausgeführt werden und mit einem Konto verknüpft sind', beträgt jetzt 1000. Die 'Maximale Anzahl von benutzerdefinierten Bereitstellungskonfigurationen, die einem AWS Konto zugeordnet sind', ist jetzt 100. Weitere Informationen zu Limits finden Sie im Thema Limits.	08. August 2022
Es wurde eine Tabelle hinzugefügt, in der die in den einzelnen Regionen unterstützten CodeDeploy Endpunkte aufgeführt sind.	Weitere Informationen finden Sie unter Verwendung CodeDeploy mit Amazon Virtual Private Cloud.	20. April 2022

Ein neues Limit für Amazon blue/green ECS-Bereitstellungen wurde hinzugefügt.	Die maximale Anzahl von Stunden zwischen der Bereitstellung einer Revision und der Verlagerung des Datenverkehrs in die Ersatzumgebung während einer Amazon blue/green ECS-Bereitstellung beträgt jetzt 120. Weitere Informationen finden Sie unter Bereitstellungen im Thema Limits.	12. April 2022
Es wurde ein Thema zur Vermeidung des Problems mit dem verwirrten Stellvertreter hinzugefügt	Weitere Informationen erhalten Sie unter AWS Identity and Access Management für AWS CodeDeploy.	14. März 2022
CodeDeploy hat eine bestehende AWS verwaltete Richtlinie aktualisiert	Die `AmazonEC2RoleforAWSCodeDeployLimited` Rolle wurde aktualisiert. Weitere Informationen finden Sie unter AWS Verwaltete Richtlinienaktualisierungen.	22. November 2021
CodeDeploy hat eine bestehende AWS verwaltete Richtlinie aktualisiert	Das `AWS CodeDeployRole` wurde aktualisiert. Weitere Informationen finden Sie unter AWS Verwaltete Richtlinienaktualisierungen.	18. Mai 2021
CodeDeploy Version v1.3.2 für Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.3.2 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	6. Mai 2021

CodeDeploy unterstützt die Aktualisierung veralteter Amazon EC2 EC2-Instances	CodeDeploy unterstützt jetzt die automatische Aktualisierung veralteter Amazon EC2 EC2-Instances. Weitere Informationen finden Sie unter Erweiterte Optionen für eine Bereitstellungsgruppe konfigurieren.	23. Februar 2021
CodeDeploy Version v1.3.1 für Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.3.1 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	22. Dezember 2020
CodeDeploy Version v1.3.0 für den Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.3.0 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	10. November 2020
CodeDeploy Version v1.2.1 für den Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.2.1 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	23. September 2020

CodeDeploy unterstützt Amazon VPC-Endpunkte, die von AWS PrivateLink	Wenn Sie Amazon Virtual Private Cloud (Amazon VPC) zum Hosten Ihrer AWS Ressourcen verwenden , können Sie eine private Verbindung zwischen Ihrer VPC und herstellen. CodeDeploy Sie können diese Verbindung verwenden, um CodeDeploy die Kommunikation mit den Ressourcen in der VPC zu ermöglichen, ohne das öffentliche Internet verwenden zu müssen. Weitere Informationen finden Sie unter Verwendung CodeDeploy mit Amazon Virtual Private Cloud.	11. August 2020
Die CodeDeploy Servicebeschränkungen wurden aktualisiert	Das Limit sowohl für die Anzahl der Anwendungen pro Konto als auch für die Anzahl der Bereitstellungsgruppen pro Anwendung wurde auf 1000 aktualisiert. Weitere Informationen zu CodeDeploy Dienstbeschränkungen finden Sie unter CodeDeploy Grenzwerte.	6. August 2020
CodeDeploy Version v1.1.2 für Agenten	Der AWS CodeDeploy Agent wurde auf Version 1.1.2 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	4. August 2020

CodeDeploy Veröffentlichung von Agent 1.1.0 und Integration mit Amazon EC2 Systems Manager	Version 1.1.0 des CodeDeploy Agenten ist jetzt verfügbar. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten. Sie können jetzt Amazon EC2 Systems Manager verwenden, um Ihre CodeDeploy Agenteninstallation und Updates auf Ihren Amazon EC2- oder lokalen Instances automatisch zu verwalten. Weitere Informationen finden Sie unter Installieren des CodeDeploy Agenten mithilfe von Amazon EC2 Systems Manager.	30. Juni 2020
CodeDeploy unterstützt die Verwaltung von Amazon blue/green ECS-Bereitstellungen mit CloudFormation	Sie können es jetzt verwenden CloudFormation, um Amazon blue/green ECS-Bereitstellungen über CodeDeploy zu verwalten. Sie generieren Ihre Bereitstellung, indem Sie Ihre grünen und blauen Ressourcen definieren und die Einstellungen für das Routing und die Stabilisierung des Datenverkehrs angeben, die in CloudFormation verwendet werden sollen. Weitere Informationen finden Sie unter Erstellen einer Amazon blue/green ECS-Bereitstellung durch CloudFormation.	19. Mai 2020

CodeDeploy unterstützt gewichtete Verkehrsverlagerung für Amazon blue/green ECS-Bereitstellungen	CodeDeploy unterstützt jetzt die gewichtete Verkehrsverlagerung für Amazon blue/green ECS-Bereitstellungen. Sie wählen oder erstellen eine Bereitstellungskonfiguration, um die Anzahl der Datenverkehrsverlagerungs-Intervalle in der Bereitstellung und den Prozentsatz des zu verlagernden Datenverkehrs in jedem Intervall anzugeben. Das folgende Thema wurde aktualisiert, um dieser Änderung Rechnung zu tragen: Bereitstellungskonfigurationen auf einer Amazon ECS-Rechenplattform.	6. Februar 2020
Die Themen Sicherheit, Authentifizierung und Zugriffskontrolle wurden aktualisiert	Die Informationen zu Sicherheit, Authentifizierung und Zugriffskontrolle für CodeDeploy wurden in einem neuen Sicherheitskapitel zusammengefasst. Weitere Informationen finden Sie unter Sicherheit.	26. November 2019
CodeDeploy unterstützt Benachrichtigungsregeln	Sie können jetzt Benachrichtigungsregeln verwenden, um Benutzer über wichtige Änderungen in Bereitstellungen zu benachrichtigen. Weitere Informationen finden Sie unter Erstellen einer Benachrichtigungsregel.	5. November 2019

Aktualisierte Themen	CodeDeploy ist jetzt in der Region Asien-Pazifik (Hongkong) (ap-east-1) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen. Sie müssen den Zugriff auf diese Region ausdrücklich aktivieren. Weitere Informationen finden Sie unter AWS Regionen verwalten.	25. April 2019

Aktualisierte Themen	AWS CodeDeploy unterstützt jetzt blue/green Bereitstellungen einer containerisierten Anwendung in einem Amazon ECS-Service. Eine CodeDeploy Anwendung, die die neue Amazon ECS-Rechenplattform verwendet, stellt eine containerisierte Anwendung für einen neuen Ersatzaufgabensatz im selben Amazon ECS-Service bereit. Um dieser Änderung Rechnung zu tragen, wurden mehrere Themen hinzugefügt und aktualisiert, darunter Überblick über AWS CodeDeploy Rechenplattformen, Bereitstellungen auf einer Amazon ECS-Rechenplattform, AppSpec Dateistruktur für Amazon ECS-Bereitstellungen und Erstellen einer Anwendung für eine Amazon ECS-Servicebereitstellung (Konsole).	27. November 2018
Der Agent wurde aktualisiert CodeDeploy	Der AWS CodeDeploy Agent wurde auf Version 1.0.1.1597 aktualisiert. Weitere Informationen finden Sie unter Versionsverlauf des Agenten. CodeDeploy	15. November 2018

Aktualisierte Konsole	Die Verfahren in diesem Handbuch wurden aktualisiert, um das neue Design der CodeDeploy-Konsole widerzuspiegeln.	30. Oktober 2018
Neue unterstützte Mindestversion des CodeDeploy Agenten	Die unterstützte Mindestversion des AWS CodeDeploy Agenten ist jetzt 1.8.x. Weitere Informationen finden Sie unter Versionsverlauf des CodeDeploy Agenten.	7. August 2018

Frühere Aktualisierungen

In der folgenden Tabelle sind wichtige Änderungen in jeder Version des AWS CodeDeploy - Benutzerhandbuchs vor Juni 2018 beschrieben.

Änderungen	Beschreibung	Änderungsdatum
Themenaktualisierungen	CodeDeploy ist jetzt in der Region Europa (Paris) (eu-west-3) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen.	19. Dezember 2017
Aktualisierte Themen	CodeDeploy ist jetzt in der Region China (Ningxia) verfügbar. Um Dienste in der Region China (Peking) oder China (Ningxia) nutzen zu können, benötigen Sie ein Konto und Anmeldeinformationen für diese Regionen. Konten und Anmeldeinformationen für andere AWS Regionen funktionieren nicht für die Regionen Peking und Ningxia und umgekehrt.	11. Dezember 2017

Änderungen	Beschreibung	Änderungsdatum
	Informationen zu einigen Ressourcen für die Regionen China, wie z. B. CodeDeploy Resource Kit-Bucket-Namen und Verfahren zur CodeDeploy Agenteninstallation, sind in dieser Ausgabe des CodeDeploy Benutzerhandbuchs nicht enthalten. Weitere Informationen: • CodeDeployin Erste Schritte AWS in der Region China (Peking) • CodeDeploy Benutzerhandbuch für die Regionen China (englische Version | chinesische Version)	

Änderungen	Beschreibung	Änderungsdatum
Aktualisierte Themen	CodeDeploy unterstützt jetzt die Bereitstellung einer Lambda-Funktion. Eine AWS Lambda -Bereitstellung verlagert eingehenden Datenverkehr von einer vorhanden en Lambda-Funktion auf eine aktualisierte Lambda-Fu nktionsversion. Sie wählen oder erstellen eine Bereitste llungskonfiguration, um die Anzahl der Intervalle zur Verkehrsverlagerung in der Bereitstellung und den Prozentsatz des Datenverkehrs, der in jedem Intervall verschoben werden soll, anzugeben. AWS Lambda Bereitstellungen werden vom AWS Serverless Applicati on Model (AWS SAM) unterstützt, sodass Sie eine AWS SAM-Bereitstellungseinstellung verwenden können, um die Art und Weise zu verwalten, wie der Datenverkehr während einer Bereitstellung verlagert wird. AWS Lambda Verschiedene Themen wurden hinzugefügt und aktualisiert, um diese Änderung widerzuspiegeln, darunter Überblick über CodeDeploy Rechenplattformen, Bereitstellungen auf einem AWS Lambda Compute-Plattform, Erstellen Sie ein AWS Bereitstellung der Lambda Compute Platform (Konsole), Erstellen Sie eine Anwendung für ein AWS Lambda Bereitstellung von Funktionen (Konsole) und Fügen Sie eine AppSpec Datei hinzu für AWS Lambda-Be reitstellung.	28. November 2017

Änderungen	Beschreibung	Änderungsdatum
Neues Thema	CodeDeploy unterstützt jetzt Bereitstellungen direkt auf einem lokalen Computer oder einer lokalen Instanz, auf der der Agent installiert ist CodeDeploy . Sie können eine Bereitstellung lokal testen und, falls sie Fehler enthält, CodeDeploy Agenten-Fehlerprotokolle verwenden, um diese zu debuggen. Sie können eine lokale Bereitstellung auch verwenden, um die Integrität einer Anwendungsrevision, den Inhalt einer AppSpec Datei und mehr zu testen. Weitere Informationen finden Sie unter Verwenden Sie den CodeDeploy Agenten, um ein Bereitstellungspaket auf einem lokalen Computer zu validieren.	16. November 2017
Aktualisierte Themen	CodeDeploy Die Unterstützung für Elastic Load Balancing Load Balancer in Bereitstellungsgruppen wurde um Network Load Balancer sowohl für blue/green Implementierungen als auch für In-Place-Bereitstellungen erweitert. Sie können jetzt einen Application Load Balancer, Classic Load Balancer oder Network Load Balancer für Ihre Bereitstellungsgruppe auswählen. Load Balancer sind für Bereitstellungen erforderlich und für blue/green direkte Bereitstellungen optional. Eine Reihe von Themen wurden aktualisiert, um diese zusätzliche Unterstützung widerzuspiegeln, einschließlich: Integrating CodeDeploy with Elastic Load Balancing, Erstellen Sie eine Anwendung für eine direkte Bereitstellung (Konsole), Voraussetzungen für die Bereitstellung, Integrating CodeDeploy with Elastic Load Balancing und Erstellen Sie eine Anwendung für eine direkte Bereitstellung (Konsole).	12. September 2017

Änderungen	Beschreibung	Änderungsdatum
Aktualisierte Themen	CodeDeploy Die Unterstützung für Elastic Load Balancing Load Balancer in Bereitstellungsgruppen wurde um Application Load Balancer sowohl für blue/green Implementierungen als auch für In-Place-Bereitstellungen erweitert. Sie können jetzt zwischen einem Application Load Balancer und einem Classic Load Balancer für Ihre Bereitstellungsgruppe wählen. Load Balancer sind für Bereitstellungen erforderlich und für blue/green direkte Bereitstellungen optional. Die Themen Integrating CodeDeploy with Elastic Load Balancing, Erstellen Sie eine Anwendung mit CodeDeploy und Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy wurden aktualisiert, um diese zusätzliche Unterstützung widerzuspiegeln.	10. August 2017

Änderungen	Beschreibung	Änderungsdatum
Neue und aktualisierte Themen	CodeDeploy unterstützt jetzt die Verwendung mehrerer Tag-Gruppen, um Unions und Überschneidungen von Instanzen zu identifizieren, die in eine Bereitstellungsgruppe aufgenommen werden sollen. Wenn Sie eine einzelne Tag-Gruppe verwenden, ist jede von mindestens einem Tag in der Gruppe identifizierte Instance in der Bereitstellungsgruppe enthalten. Wenn Sie mehrere Tag-Gruppen verwenden, sind nur die Instances, die durch mindestens einen Tag in jeder Tag-Gruppe identifiziert werden, enthalten. Weitere Informationen über die neue Methode für das Hinzufügen von Instances zu einer Bereitstellungsgruppe finden Sie unter Tagging Instances for Deployments. Weitere für diese Unterstützung aktualisierte Themen sind u. a. Erstellen Sie eine Anwendung für eine direkte Bereitstellung (Konsole), Erstellen Sie eine Anwendung für eine blue/green Bereitstellung (Konsole), Erstellen Sie eine Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole), Erstellen Sie eine Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung (Konsole), Deployments und Schritt 5: Erstellen Sie eine Anwendung und eine Bereitstellungsgruppe in Tutorial: Verwenden Sie CodeDeploy zum Bereitstellen einer Anwendung von GitHub.	31. Juli 2017

Änderungen	Beschreibung	Änderungsdatum
Aktualisiertes Thema	Zwei zusätzliche Methoden zur Installation des CodeDeploy Agenten auf Windows Server-Instanzen wurden hinzugefügtInstallieren Sie den CodeDeploy Agenten für Windows Server. Zusätzlich zu den PowerShell Windows-Befehlen sind jetzt auch Anweisungen zum Herunterladen der Installationsdatei über einen direkten HTTPS-Link und mithilfe eines Amazon S3 S3-Kopierbefehls verfügbar. Nachdem die Datei heruntergeladen oder zu einer Instance kopiert wurde, können Sie die Installation manuell ausführen.	12. Juli 2017
Aktualisierte Themen	CodeDeploy hat die Verwaltung von Verbindungen zu GitHub Konten und Repositorys verbessert. Sie können jetzt bis zu 25 Verbindungen zu GitHub Konten erstellen und speichern, um CodeDeploy Anwendungen mit GitHub Repositorys zu verknüpfen. Jede Verbindung kann mehrere Repositorys unterstützen. Sie können Verbindungen zu bis zu 25 verschiedenen GitHub Konten oder mehr als eine Verbindung zu einem einzelnen Konto erstellen . CodeDeploy Verwaltet die erforderlichen Zugriffsberechtigungen, nachdem Sie eine Anwendung mit einem GitHub Konto verbunden haben, ohne dass Sie weitere Maßnahmen ergreifen müssen. Aktualisierungen in Bezug auf diese Unterstützung wurden unter Geben Sie Informationen zu einer Revision an, die in einem GitHub Repository gespeichert ist, Integrieren CodeDeploy mit GitHub und Tutorial: Verwenden Sie CodeDeploy zum Bereitstellen einer Anwendung von GitHub vorgenommen.	30. Mai 2017

Änderungen	Beschreibung	Änderungsdatum
Aktualisierte Themen	Wenn der CodeDeploy Agent in der Vergangenheit Dateien an einem Zielort entdeckte, die nicht Teil der Anwendungsrevision der letzten erfolgreichen Bereitstellung waren, schlug er standardmäßig bei der aktuellen Bereitstellung fehl. CodeDeploy bietet jetzt Optionen dafür, wie der Agent mit diesen Dateien umgeht: Fehler bei der Bereitstellung, Beibehaltung des Inhalts oder Überschreiben des Inhalts. Erstellen Sie eine Bereitstellung mit CodeDeploywurde aktualisiert, um dieser Unterstützung Rechnung zu tragen, und der neue Abschnitt Rollback-Verhalten bei vorhandenem Inhalt wurde hinzugefügtEine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy.	16. Mai 2017
Aktualisierte Themen	Ein Classic Load Balancer in Elastic Load Balancing kann jetzt mithilfe der CodeDeploy Konsole oder AWS CLI einer Bereitstellungsgruppe zugewiesen werden. Während einer In-Situ-Bereitstellung verhindert ein Load Balancer, dass Internetdatenverkehr an eine Instance weitergeleitet wird, solange die Bereitstellung ausgeführt wird, und macht die Instance anschließend, wenn die Bereitstellung für sie abgeschlossen wurde, wieder für den Datenverkehr verfügbar. Es wurden mehrere Themen aktualisiert, um diese neue Unterstützung widerzuspiegeln, darunter Integration mit anderen AWS service, Integrating CodeDeploy with Elastic Load Balancing, Erstellen Sie eine Anwendung für eine direkte Bereitstellung (Konsole), Erstellen Sie eine Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole) und AppSpec Abschnitt „Hooks“. Einer neuer Abschnitt Behebung eines Fehlers oder eines ApplicationStop BeforeBlockTraffic Ereignisses im Lebenszyklus einer AfterBlockTraffic Bereitstellung wurde zum Fehlerbehebungshandbuch hinzugefügt.	27. April 2017

Änderungen	Beschreibung	Änderungsdatum
Aktualisierte Themen	Ein Classic Load Balancer in Elastic Load Balancing kann jetzt mithilfe der CodeDeploy Konsole oder AWS CLI einer Bereitstellungsgruppe zugewiesen werden. Während einer In-Situ-Bereitstellung verhindert ein Load Balancer, dass Internetdatenverkehr an eine Instance weitergeleitet wird, solange die Bereitstellung ausgeführt wird, und macht die Instance anschließend, wenn die Bereitstellung für sie abgeschlossen wurde, wieder für den Datenverkehr verfügbar. Es wurden mehrere Themen aktualisiert, um diese neue Unterstützung widerzuspiegeln, darunter Integration mit anderen AWS service, Integrating CodeDeploy with Elastic Load Balancing, Erstellen Sie eine Anwendung für eine direkte Bereitstellung (Konsole), Erstellen Sie eine Bereitstellungsgruppe für eine direkte Bereitstellung (Konsole) und AppSpec Abschnitt „Hooks“. Einer neuer Abschnitt Behebung eines Fehlers oder eines ApplicationStop BeforeBlockTraffic Ereignisses im Lebenszyklus einer AfterBlockTraffic Bereitstellung wurde zum Fehlerbehebungshandbuch hinzugefügt.	1. Mai 2017

Änderungen	Beschreibung	Änderungsdatum
Aktualisierte Themen	CodeDeploy ist jetzt in der Region China (Peking) verfügbar. Um Dienste in der Region China (Peking) oder China (Ningxia) nutzen zu können, benötigen Sie ein Konto und Anmeldeinformationen für diese Regionen. Konten und Anmeldeinformationen für andere AWS Regionen funktionieren nicht für die Regionen Peking und Ningxia und umgekehrt. Informationen zu einigen Ressourcen für die Regionen China, wie z. B. CodeDeploy Resource Kit-Bucket-Namen und Verfahren zur CodeDeploy Agenteninstallation, sind in dieser Ausgabe des CodeDeploy Benutzerhandbuchs nicht enthalten. Weitere Informationen: • CodeDeployin Erste Schritte AWS in der Region China (Peking) • CodeDeploy Benutzerhandbuch für die Regionen China (englische Version | chinesische Version)	29. März 2017

Änderungen	Beschreibung	Änderungsdatum
Neue und aktualisierte Themen	Es wurden mehrere neue Themen eingeführt, um der neuen CodeDeploy Unterstützung für blue/green Bereitstellungen Rechnung zu tragen. Dabei handelt es sich um eine Bereitstellungsmethode, bei der die Instanzen in einer Bereitstellungsgruppe (der ursprünglichen Umgebung) durch eine andere Gruppe von Instanzen (die Ersatzumgebung) ersetzt werden. Überblick über eine Bereitstellung blue/greenbietet eine ausführliche Erläuterung der von CodeDeploy verwendeten blue/green Methodik. Zu den zusätzlichen neuen Themen gehören Erstellen Sie eine Anwendung für eine blue/green Bereitstellung (Konsole). Erstellen Sie eine Bereitstellungsgruppe für eine EC2/On-Premises blue/green Bereitstellung (Konsole) und Richten Sie in Elastic Load Balancing einen Load Balancer für CodeDeploy Amazon EC2 EC2-Bereitstellungen ein. Zahlreiche Themen wurden aktualisiert, z. B. Erstellen Sie eine Bereitstellung mit CodeDeploy, Arbeiten mit Bereitstellungskonfigurationen in CodeDeploy, Erstellen Sie eine Anwendung mit CodeDeploy, Arbeiten mit Bereitstellungsgruppen in CodeDeploy , Arbeiten mit Bereitstellungen in CodeDeploy und AppSpec Abschnitt „Hooks“.	25. Januar 2017
Neue und aktualisierte Themen	Ein neues Thema,, beschreibtVerwenden Sie den Befehl register-on-premises-instance (IAM Session ARN), um eine lokale Instanz zu registrieren, wie lokale Instances mithilfe regelmäßig aktualisierter temporärer Anmeldeinformationen authentifiziert und registriert werden, die durch generiert werden. AWS -Security-Token-Service Dieser Ansatz ist hilfreicher für die Unterstützung einer großen Menge lokaler Instances als nur die statischen IAM-Benutzer-Anmeldeinformationen für jede Instance. Working with On-Premises Instances wurde ebenfalls aktualisiert, um diese neue Unterstützung widerzuspiegeln.	28. Dezember 2016

Änderungen	Beschreibung	Änderungsdatum
Aktualisierte Themen	CodeDeploy ist jetzt in der Region Europa (London) (eu-west-2) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen.	13. Dezember 2016
Aktualisierte Themen	CodeDeploy ist jetzt in der Region Kanada (Central) (ca-central-1) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen.	8. Dezember 2016
Aktualisierte Themen	CodeDeploy ist jetzt in der Region USA Ost (Ohio) (us-east-2) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen.	17. Oktober 2016
Neue Themen	Ein neuer Abschnitt, Authentifizierung und Zugriffskontrolle, bietet umfassende Informationen zur Verwendung von AWS Identity and Access Management (IAM) und CodeDeploy zur Sicherung des Zugriffs auf Ihre Ressourcen mithilfe von Anmeldeinformationen. Diese Anmeldeinformationen bieten die erforderlichen Berechtigungen für den Zugriff auf AWS Ressourcen, z. B. das Abrufen von Anwendungsrevisionen aus Amazon S3 S3-Buckets und das Lesen der Tags auf Amazon EC2 EC2-Instances.	11. Oktober 2016

Änderungen	Beschreibung	Änderungsdatum
Aktualisiertes Thema	Aktualisieren Sie den CodeDeploy Agenten auf Windows Serverwurde aktualisiert, um der Verfügbarkeit des neuen CodeDeploy Agenten-Updaters für Windows Server Rechnung zu tragen. Bei der Installation auf einer Windows Server-Instanz sucht der Updater regelmäßig nach neuen Versionen. Wenn der Updater eine neue Version findet, deinstalliert er die aktuelle Agent-Version, sofern eine installiert ist, bevor er die neueste Version installiert.	4. Oktober 2016
Aktualisierte Themen	CodeDeploy ist jetzt in CloudWatch Amazon-Alarme integriert, sodass es möglich ist, eine Bereitstellung zu beenden, wenn sich der Status eines bestimmten Alarms für mehrere aufeinanderfolgende Zeiträume ändert, wie im Alarmschwellenwert angegeben. CodeDeploy unterstützt jetzt auch das automatische Zurücksetzen einer Bereitstellung, wenn bestimmte Bedingungen erfüllt sind, z. B. ein Bereitstellungsfehler oder ein aktivierter Alarm. Eine Reihe von Themen wurden aktualisiert, um diese Änderungen widerzuspiegeln, einschließlich Erstellen Sie eine Anwendung mit CodeDeploy, Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy, Ändern Sie die Einstellungen der Bereitstellungsgruppe mit CodeDeploy, Deployments, Eine Bereitstellung erneut bereitstellen und rückgängig machen mit CodeDeploy und Produkt- und Serviceintegrationen mit CodeDeploy sowie ein neues Thema: Überwachung von Bereitstellungen mit CloudWatch Alarmen in CodeDeploy.	15. September 2016

Änderungen	Beschreibung	Änderungsdatum
Neue und aktualisierte Themen	CodeDeploy bietet jetzt die Integration mit Amazon CloudWatch Events. Sie können CloudWatch Ereignisse jetzt verwenden, um eine oder mehrere Aktionen einzuleiten, wenn Änderungen am Status einer Bereitstellung oder am Status einer Instance, die zu einer CodeDeploy Bereitstellungsgruppe gehört, erkannt werden. Sie können Aktionen integrieren, die AWS Lambda Funktionen aufrufen, die in Kinesis-Streams oder Amazon SNS SNS-Themen veröffentlichen, Nachrichten an Amazon SQS SQS-Warteschlangen weiterleiten oder die wiederum Alarmaktionen auslösen. CloudWatch Weitere Informationen finden Sie unter Überwachung von Bereitstellungen mit Amazon Events CloudWatch.	9. September 2016
Themenaktualisierungen	Die Themen A Integrating CodeDeploy with Elastic Load Balancing und B Integration mit anderen AWS service wurden aktualisiert, um eine zusätzliche Option für den Lastenausgleich zu berücksichtigen. CodeDeploy unterstützt jetzt den Classic Load Balancer und den Application Load Balancer, die in Elastic Load Balancing verfügbar sind.	11. August 2016
Themenaktualisierungen	CodeDeploy ist jetzt in der Region Asien-Pazifik (Mumbai) (ap-south-1) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen.	27. Juni 2016

Änderungen	Beschreibung	Änderungsdatum
Themenaktualisierungen	CodeDeploy ist jetzt in der Region Asien-Pazifik (Seoul) (ap-northeast-2) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen. Das Inhaltsverzeichnis wurde neu angeordnet und enthält nun Abschnitte für Instances, Bereitstellungskonfigurationen, Anwendungen, Bereitstellungsgruppen, Revisionen und Bereitstellungen. CodeDeploy-Tutorials wurde ein neuer Abschnitt hinzugefügt. Für eine bessere Benutzerfreundlichkeit wurden einige lange Themen, einschließlich CodeDeploy AppSpec Dateiverweis und Problembehebung CodeDeploy, in kürzere Themen unterteilt. Die Konfigurationsinformationen für den CodeDeploy-Agenten wurden in ein neues Thema verschoben: CodeDeploy Referenz zur Agentenkonfiguration.	15. Juni 2016
Neue und aktualisierte Themen	Fehlercodes für AWS CodeDeploy bietet Informationen über einige der Fehlermeldungen, die ggf. angezeigt werden, wenn CodeDeploy -Bereitstellungen fehlschlagen. Die folgenden Abschnitte in Problembehebung CodeDeploy wurden aktualisiert, um die Lösung von Bereitstellungsproblemen besser zu unterstützen: • EC2-Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe können nicht gestartet werden und erhalten den Fehler „Heartbeat Timeout“ • Vermeiden Sie es, mehrere Bereitstellungsgruppen einer einzigen Amazon EC2 Auto Scaling-Gruppe zuzuordnen	20. April 2016

Änderungen	Beschreibung	Änderungsdatum
Themenaktualisierungen	CodeDeploy ist jetzt in der Region Südamerika (São Paulo) (sa-east-1) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen. Mit dem CodeDeploy Agenten arbeitenwurde aktualisiert und berücksichtigt nun die neue Konfigurationsoption:max_revisions:, mit der Sie die Anzahl der Anwendungsrevisionen für eine Bereitstellungsgruppe angeben, die der Agent archivieren soll. CodeDeploy	10. März 2016

Änderungen	Beschreibung	Änderungsdatum
Neue und aktualisierte Themen	CodeDeploy unterstützt jetzt das Hinzufügen von Triggern zu einer Bereitstellungsgruppe, um Benachrichtigungen über Ereignisse im Zusammenhang mit Bereitstellungen oder Instanzen in dieser Bereitstellungsgruppe zu erhalten. Diese Benachrichtigungen werden an Empfänger gesendet, die ein Amazon Simple Notification Service-Thema abonniert haben, das Sie in die Aktion des Auslösers aufgenommen haben. Sie können auch JSON-Daten verwenden, die erstellt werden, wenn ein Trigger in Ihrem eigenen angepassten Benachrichtigungs-Workflow ausgelöst wird. Weitere Informationen finden Sie unter Monitoring Deployments with Amazon SNS Event Notifications. Verfahren wurden aktualisiert, um die Umgestaltung der Seite mit den Anwendungsdetails widerzuspiegeln. Der Abschnitt Bereitstellungen schlagen für bis zu einer Stunde nicht fehl, nachdem eine Instance während einer Bereitstellung beendet wurde in Problembehebung CodeDeploy wurde aktualisiert. CodeDeploy Kontingentewurde aktualisiert und berücksichtigt nun die geänderten Grenzwerte für die Anzahl der Bereitstellungsgruppen, die einer einzelnen Anwendung zugeordnet werden können, die zulässigen Werte für die Mindesteinstellungen für fehlerfreie Instances und die AWS SDK für Ruby erforderlichen Versionen von.	17. Februar 2016

Änderungen	Beschreibung	Änderungsdatum
Neue und aktualisierte Themen	CodeDeploy ist jetzt in der Region USA West (Nordkalifornien) (us-west-1) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Hinzufügung dieser neuen Region Rechnung zu tragen. Unter Wählen Sie einen CodeDeploy Repository-Typ sind die Repository-Typen aufgelistet und beschrieben, die CodeDeploy jetzt unterstützt. Dieses neue Thema wird aktualisiert, um Unterstützung bei der Einführung anderer Repository-Typen zu bieten. Verwaltung des CodeDeploy Agentenbetriebswurde mit Informationen über die neue `.version` Datei aktualisiert, die den Instanzen hinzugefügt wurde, um die aktuelle Version des CodeDeploy Agenten zu melden, sowie Informationen zu unterstützten Versionen des Agenten. Syntax-Hervorhebung für Codebeispiele, einschließlich JSON- und YAML-Beispiele, finden Sie nun im Benutzerhandbuch. Hinzufügen einer Anwendungsspezifikationsdatei zu einer Revision für CodeDeploy wurde als Schritt-für-Schritt-Anweisung umstrukturiert.	20. Januar 2016

Änderungen	Beschreibung	Änderungsdatum
Neues Thema	Stellen Sie eine Anwendung in einer anderen Umgebung bereit AWS Kontobeschreibt die Setup-Anforderungen und den Prozess für die Initiierung von Bereitstellungen, die zu einem anderen Konto Ihres Unternehmens gehören, ohne dass vollständige Anmeldeinformationen für dieses andere Konto erforderlich sind. Dies ist besonders nützlich für Unternehmen, die mehrere Konten für unterschiedliche Zwecke verwenden, z. B. eins für Entwicklungs- und Testumgebungen und ein anderes im Zusammenhang mit der Produktionsumgebung.	30. Dezember 2015
Aktualisiertes Thema	Das Thema Produkt- und Serviceintegrationen mit CodeDeploy wurde überarbeitet. Es enthält jetzt einen Abschnitt für die Integration von Beispielen aus der Community, mit Listen von Blog-Beiträgen und Videobeispielen in Bezug auf CodeDeploy-Integrationen.	16. Dezember 2015
Themenaktualisierungen	CodeDeploy ist jetzt in der Region Asien-Pazifik (Singapur) (ap-southeast-1) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen.	9. Dezember 2015
Themenaktualisierungen	Mit dem CodeDeploy Agenten arbeiten wurde aktualisiert, um die neue Option `:proxy_uri:` in der CodeDeploy-Agentenkonfigurationsdatei einzubeziehen. CodeDeploy AppSpec Dateiverweis wurde mit Informationen über die Verwendung einer neuen Umgebungsvariable `DEPLOYMENT_GROUP_ID` aktualisiert, auf die Hook-Skripts während eines Ereignisses im Bereitstellungslebenszyklus zugreifen können.	1. Dezember 2015

Änderungen	Beschreibung	Änderungsdatum
Aktualisiertes Thema	Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploywurde aktualisiert, um dem neuen Verfahren für die Erstellung einer Servicerolle Rechnung zu tragen CodeDeploy und weitere Verbesserungen zu berücksichtigen.	13. November 2015
Themenaktualisierungen	CodeDeploy ist jetzt in der Region Europa (Frankfurt) (eu-central-1) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen. Das Thema Problembehebung CodeDeploy wurde mit Informationen aktualisiert, um sicherzustellen, dass die Zeiteinstellungen auf Instances korrekt sind.	19. Oktober 2015
Neue Themen	CloudFormation Vorlagen als CodeDeploy Referenzwurde veröffentlicht, um der neuen CloudFormation Unterstützung von CodeDeploy Maßnahmen Rechnung zu tragen. Das Thema Primary Components wurde erstellt und die Definition einer Zielrevision eingeführt.	1. Oktober 2015
Themenaktualisierungen	Erstellen Sie eine Bereitstellungsgruppe mit CodeDeploy wurde aktualisiert und bietet jetzt die Möglichkeit, Instances für eine Bereitstellungsgruppe mit Platzhalter-Suchvorgängen zu suchen. Instance Health wurde mit Informationen zum Konzept der Mindestanzahl fehlerfreier Instances aktualisiert.	31. August 2015

Änderungen	Beschreibung	Änderungsdatum
Themenaktualisierungen	CodeDeploy ist jetzt in der Region Asien-Pazifik (Tokio) (ap-northeast-1) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Region Rechnung zu tragen.	19. August 2015
Themenaktualisierungen	CodeDeploy unterstützt jetzt Bereitstellungen auf lokalen Red Hat Enterprise Linux (RHEL) -Instances und Amazon EC2 EC2-Instances. Weitere Informationen finden Sie unter den folgenden Themen: • Betriebssysteme, die vom Agenten unterstützt werden CodeDeploy • Arbeiten mit Instanzen für CodeDeploy • Tutorial: Bereitstellung WordPress auf einer Amazon EC2 EC2-Instance (Amazon Linux oder Red Hat Enterprise Linux und Linux, macOS oder Unix) • Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat Enterprise Linux)	23. Juni 2015
Aktualisiertes Thema	CodeDeploy stellt jetzt eine Reihe von Umgebungsvariablen bereit, die Ihre Bereitstellungsskripts bei Bereitstellungen verwenden können. Zu diesen Umgebungsvariablen gehören Informationen wie der Name der aktuellen CodeDeploy Anwendung, der Bereitstellungsgruppe und des Ereignisses im Bereitstellungslebenszyklus sowie die aktuelle CodeDeploy Bereitstellungs-ID. Weitere Informationen finden Sie am Ende des Abschnitts AppSpec Abschnitt „Hooks“ in der CodeDeploy AppSpec Dateiverweis.	29. Mai 2015

Änderungen	Beschreibung	Änderungsdatum
Themenaktualisierungen	CodeDeploy stellt jetzt eine Reihe AWS verwalteter Richtlinien in IAM bereit, die Sie verwenden können, anstatt die entsprechenden Richtlinien manuell selbst zu erstellen. Dazu zählen: • Eine Richtlinie, die es einem Benutzer ermöglicht, Revisionen CodeDeploy nur bei Only zu registrieren und sie dann bereitzustellen. CodeDeploy • Eine Richtlinie, die einem Benutzer vollen Zugriff auf CodeDeploy Ressourcen gewährt. • Eine Richtlinie, mit der einem Benutzer nur Lesezugriff auf Ressourcen gewährt wird. CodeDeploy • Eine Richtlinie, die an eine Servicerolle angehängt wird, CodeDeploy sodass Amazon EC2-Instances anhand ihrer Amazon EC2-Tags, lokalen Instance-Tags oder Amazon EC2 Auto Scaling-Gruppennamen identifiziert und entsprechende Anwendungsrevisionen für sie bereitgestellt werden können. Weitere Informationen finden Sie im Abschnitt Customer-managed Beispiele für politische Maßnahmen unter "Authentifizierung und Zugriffskontrolle".	29. Mai 2015
Themenaktualisierungen	CodeDeploy ist jetzt in der Region Europa (Irland) (eu-west-1) und der Region Asien-Pazifik (Sydney) (ap-southeast-2) verfügbar. Verschiedene Themen, darunter auch die mit Anweisungen zur Einrichtung des CodeDeploy Agenten, wurden aktualisiert, um der Verfügbarkeit dieser neuen Regionen Rechnung zu tragen.	7. Mai 2015

Änderungen	Beschreibung	Änderungsdatum
Neue Themen	CodeDeploy unterstützt jetzt Bereitstellungen für lokale Instances und Amazon EC2 EC2-Instances. Die folgenden Themen wurden hinzugefügt, um diese neue Unterstützung zu beschreiben: • Working with On-Premises Instances • Tutorial: Bereitstellen einer Anwendung auf einer lokalen Instanz mit CodeDeploy (Windows Server, Ubuntu Server oder Red Hat Enterprise Linux) • Working with On-Premises Instances	2. April 2015
Neues Thema	CodeDeploy Ressourcen wurde hinzugefügt.	2. April 2015
Aktualisiertes Thema	Problembehebung CodeDeploy wurde aktualisiert: • Ein neuer Abschnitt Long-running Prozesse können dazu führen, dass Bereitstellungen fehlschlagen beschreibt die Schritte zum Identifizieren und Beheben von Bereitste llungsausfällen durch lange laufende Prozesse. • Der Allgemeine Amazon EC2 Auto Scaling Scaling-F ehlerbehebung Abschnitt wurde aktualisiert und zeigt, dass die CodeDeploy Amazon EC2 Auto Scaling-T imeout-Logik für den CodeDeploy Agenten von fünf Minuten auf eine Stunde erhöht wurde. • In einem neuen Nicht übereinstimmende Amazon EC2 Auto Scaling Scaling-Lifecycle-Hooks können dazu führen, dass automatische Bereitstellungen für Amazon EC2 Auto Scaling Scaling-Gruppen gestoppt werden oder fehlschlagen Abschnitt werden Schritte beschrieben, die Sie ergreifen können, um fehlgesch lagene automatische Bereitstellungen für Amazon EC2 Auto Scaling Scaling-Gruppen zu identifizieren und zu beheben.	2. April 2015

Änderungen	Beschreibung	Änderungsdatum
Themenaktualisierungen	Die folgenden Themen wurden aktualisiert, um neuen Empfehlungen für die Erstellung eigener benutzerdefinierter Richtlinien und deren anschließendes Anhängen dieser Richtlinien an Benutzer und Rollen in IAM Rechnung zu tragen: • Konfigurieren Sie eine Amazon EC2 EC2-Instance, mit der Sie arbeiten können CodeDeploy • Schritt 4: Erstellen Sie ein IAM-Instance-Profil für Ihre Amazon EC2 EC2-Instances • Schritt 2: Erstellen Sie eine Servicerolle für CodeDeploy Zwei Abschnitte wurden zu Problembehebung CodeDeploy hinzugefügt: • Allgemeine Checkliste für die Fehlerbehebung • PowerShell Windows-Skripts verwenden standardmäßig nicht die 64-Bit-Version von Windows PowerShell Der Abschnitt AppSpec Abschnitt „Hooks“ in der CodeDeploy AppSpec Dateiverweis wurde aktualisiert und beschreibt nun genauer die verfügbaren Bereitstellungslebenszyklusereignisse.	12. Februar 2015
Themenaktualisierungen	Ein neuer Abschnitt wurde zu Problembehebung CodeDeploy hinzugefügt: EC2-Instances in einer Amazon EC2 Auto Scaling Scaling-Gruppe können nicht gestartet werden und erhalten den Fehler „Heartbeat Timeout“. Ein CloudBees Abschnitt wurde hinzugefügt. Produkt- und Serviceintegrationen mit CodeDeploy	28. Januar 2015

Änderungen	Beschreibung	Änderungsdatum
Themenaktualisierungen	Die folgenden Abschnitte wurden zu Problembehebung CodeDeploy hinzugefügt: • Die Verwendung einiger Texteditoren zum Erstellen von AppSpec Dateien und Shell-Skripts kann dazu führen, dass Bereitstellungen fehlschlagen • Verwenden von Finder in macOS zur Bündelung einer Anwendungsrevision kann zum Fehlschlagen von Bereitstellungen führen • Behebung eines Fehlers oder eines ApplicationStop BeforeBlockTraffic Ereignisses im Lebenszyklus einer AfterBlockTraffic Bereitstellung • Behebung eines fehlgeschlagenen DownloadBundle Deployment-Lifecycle-Ereignisses mit UnknownError: nicht zum Lesen geöffnet • Allgemeine Amazon EC2 Auto Scaling Scaling-Fehlerbehebung	20. Januar 2015
Neue Themen	Der Abschnitt Produkt- und Serviceintegrationen mit CodeDeploy wurde mit den folgenden Themen aktualisiert: • Integration CodeDeploy mit Amazon EC2 Auto Scaling • Tutorial: Wird verwendet CodeDeploy , um eine Anwendung für eine Auto Scaling Scaling-Gruppe bereitzustellen • Monitoring Deployments • Integrating CodeDeploy with Elastic Load Balancing • Integrieren CodeDeploy mit GitHub • Tutorial: Verwenden Sie CodeDeploy zum Bereitstellen einer Anwendung von GitHub	9. Januar 2015

Änderungen	Beschreibung	Änderungsdatum
Themenaktualisierungen	• Der Abschnitt Automatisch bereitstellen von CodePipeline mit CodeDeploy wurde zu Integrieren CodeDeploy mit GitHub hinzugefügt. Sie können jetzt automatisch ein Deployment von einem GitHub Repository aus auslösen, wenn der Quellcode in diesem Repository geändert wird. • Der Abschnitt Probleme mit Amazon EC2 Auto Scaling beheben wurde zu Problembehebung CodeDeploy hinzugefügt. In diesem neuen Abschnitt wird beschrieben, wie Sie häufig auftretende Probleme bei der Bereitstellung in Amazon EC2 Auto Scaling Scaling-Gruppen beheben können. • Der neue Unterabschnitt "Dateibeispiele" wurde zum Abschnitt AppSpec Abschnitt „Dateien“ (nur EC2/On-Premises Bereitstellungen) von CodeDeploy AppSpec Dateiverweis hinzugefügt. Dieser neue Unterabschnitt enthält mehrere Beispiele dafür, wie der `files` Abschnitt einer AppSpec Datei verwendet werden kann, um anzuweisen CodeDeploy , während einer Bereitstellung bestimmte Dateien oder Ordner an bestimmte Speicherorte auf einer Amazon EC2 EC2-Instance zu kopieren.	8. Januar 2015
Neues Thema	Monitoring Deploymentswurde hinzugefügt. CodeDeploy ist CodeDeploy in einen Service integriert AWS CloudTrail, der API-Aufrufe erfasst, die von oder im Namen Ihres AWS Kontos getätigt wurden, und die Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket übermittelt.	17. Dezember 2014
Erste veröffentlichte Version	Dies ist die erste veröffentlichte Version des CodeDeploy - Benutzerhandbuchs.	12. November 2014

AWS Glossar

Die neueste AWS Terminologie finden Sie im AWS Glossar in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.