Guida per l'utente

AWS CodeDeploy



Versione API 2014-10-06

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

AWS CodeDeploy: Guida per l’utente

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

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in qualsiasi modo che possa causare confusione tra i clienti o in qualsiasi modo che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Che cos'è CodeDeploy? 1
Vantaggi di AWS CodeDeploy 2
Panoramica delle piattaforme di elaborazione CodeDeploy 3
Panoramica dei tipi di implementazione CodeDeploy 10
Panoramica di una distribuzione sul posto 12
Panoramica di una distribuzione blue/green 13
Ci piacerebbe conoscere la tua opinione 17
Componenti principali 18
Applicazione 18
Compute platform (Piattaforma di calcolo) 18
Configurazione dell'implementazione 19
Gruppo di distribuzione 20
Il tipo di implementazione 20
Il tipo di implementazione 22
Revision 22
Ruolo del servizio 22
Revisione dell'obiettivo 23
Altri componenti 23
Distribuzioni 23
Implementazioni su un AWS Piattaforma di elaborazione Lambda 24
Implementazioni su una piattaforma di calcolo Amazon ECS 27
Distribuzioni su una piattaforma di elaborazione EC2/On-Premises 39
file con le specifiche dell'applicazione 46
AppSpec file su una piattaforma di elaborazione Amazon ECS 46
AppSpec file su un AWS Lambda piattaforma di calcolo 47
AppSpec file su una piattaforma di elaborazione EC2/on-premises 47
In che modo l' CodeDeploy agente utilizza il file AppSpec 48
Nozioni di base 49
Fase 1: Configurazione 49
Iscriviti per un Account AWS 49
Fase 2: Creare un ruolo di servizio 50
Crea un ruolo di servizio (console) 52
Creare un ruolo di servizio (CLI) 55
Ottieni il ruolo di servizio ARN (console) 58

Ottieni il ruolo di servizio ARN (CLI) 58
Passaggio 3: Limita le autorizzazioni CodeDeploy dell'utente 58
Fase 4: creare un profilo di istanza IAM 62
Crea un profilo di istanza IAM per le tue istanze Amazon EC2 (CLI) 63
Crea un profilo di istanza IAM per le tue istanze Amazon EC2 (console) 68
Integrazioni di prodotti e servizi 71
Integrazione con altri AWS servizi 71
Amazon EC2 Auto Scaling 79
Elastic Load Balancing 88
Integrazione con prodotti e servizi dei partner 92
GitHub 98
Esempi di integrazione dalla community 102
Post del blog 102
Esercitazioni 104
Tutorial: Esegui la distribuzione WordPress su un'istanza non Windows 104
Fase 1: avvio di un'istanza Amazon EC2 105
Fase 2: configura il contenuto di origine 108
Passaggio 3: carica l'applicazione su Amazon S3 113
Fase 4: Implementazione dell'applicazione 118
Fase 5: Aggiornare e ridistribuire l'applicazione 124
Fase 6: pulizia 128
Tutorial: distribuire un'applicazione Hello World su un'istanza di Windows Server 132
Fase 1: avvio di un'istanza Amazon EC2 133
Passaggio 2: configura il contenuto di origine 135
Passaggio 3: carica l'applicazione su Amazon S3 138
Fase 4: Implementa l'applicazione 143
Fase 5: Aggiorna e ridistribuisci l'applicazione 149
Fase 6: pulizia 152
Tutorial: distribuisci un'applicazione su un'istanza locale 155
Prerequisiti 156
Passaggio 1: configurare l'istanza locale 156
Passaggio 2: creare una revisione di esempio dell'applicazione 157
Fase 3: raggruppa e carica la revisione dell'applicazione su Amazon S3 162
Fase 4: Implementa la revisione dell'applicazione 162
Passaggio 5: verifica la distribuzione 162
Fase 6: eliminare le risorse 163

Tutorial: Implementazione in un gruppo di Auto Scaling 165
Prerequisiti 165
Fase 1: Creare e configurare il gruppo Auto Scaling 166
Fase 2: Distribuire l'applicazione nel gruppo Auto Scaling 173
Passaggio 3: verifica i risultati 183
Fase 4: Aumentare il numero di istanze Amazon EC2 nel gruppo Auto Scaling 185
Fase 5: Controlla nuovamente i risultati 186
Fase 6: pulizia 189
Tutorial: distribuisci un'applicazione da GitHub 191
Prerequisiti 192
Fase 1: Configura un account GitHub 192
Passaggio 2: crea un GitHub repository 192
Passaggio 3: carica un'applicazione di esempio nel tuo GitHub repository 195
Fase 4: Eseguire il provisioning di un'istanza 200
Passaggio 5: creare un'applicazione e un gruppo di distribuzione 200
Fase 6: Distribuire l'applicazione sull'istanza 203
Fase 7: Monitoraggio e verifica della distribuzione 207
Fase 8: Pulizia 208
Tutorial: distribuire un'applicazione in Amazon ECS 210
Prerequisiti 212
Fase 1: aggiorna la tua applicazione Amazon ECS 213
Fase 2: Creare il file AppSpec 214
Passaggio 3: utilizza la CodeDeploy console per distribuire l'applicazione 215
Fase 4: pulizia 219
Tutorial: distribuisci un servizio Amazon ECS con un test di convalida 220
Prerequisiti 223
Fase 1: Creare un listener di test 223
Fase 2: aggiorna la tua applicazione Amazon ECS 223
Fase 3: Creare una funzione Lambda dell'hook del ciclo di vita 224
Passaggio 4: aggiorna il file AppSpec 227
Fase 5: utilizza la CodeDeploy console per distribuire il servizio Amazon ECS 228
Fase 6: Visualizza l'output della funzione Lambda hook in Logs CloudWatch 230
Fase 7: pulire 231
Tutorial: Implementazione di una funzione Lambda utilizzando SAM AWS 232
Prerequisiti 233
Fase 1: Configura la tua infrastruttura 233

Fase 2: Aggiornare la funzione Lambda ... 248
Fase 3: Implementazione della funzione Lambda aggiornata ... 251
Fase 4: Visualizza i risultati della distribuzione ... 253
Fase 5: rimozione ... 256
Collaborare con l' CodeDeploy agente ... 258
Sistemi operativi supportati dall'agente CodeDeploy ... 259
Sistemi operativi AMI Amazon EC2 supportati ... 259
Sistemi operativi locali supportati ... 259
Protocollo di comunicazione e porta per l'agente CodeDeploy ... 260
Cronologia delle versioni dell'agente CodeDeploy ... 261
Gestione del processo CodeDeploy ... 279
Revisione dell'applicazione e pulizia dei file di registro ... 280
File installati dall'agente CodeDeploy ... 281
Gestione delle operazioni degli CodeDeploy agenti ... 284
Verifica che l' CodeDeploy agente sia in esecuzione ... 285
Determina la versione dell'agente CodeDeploy ... 288
Installa l' CodeDeploy agente ... 291
Aggiorna l' CodeDeploy agente ... 304
Disinstalla l' CodeDeploy agente ... 307
Invia i log degli CodeDeploy agenti a CloudWatch ... 308
Utilizzo delle istanze ... 314
Confronto tra istanze Amazon EC2 e istanze locali ... 314
Attività di istanza per CodeDeploy ... 316
Etichettatura delle istanze per le distribuzioni CodeDeploy ... 317
Esempio 1: gruppo di tag singolo, tag singolo ... 318
Esempio 2: gruppo di tag singolo, tag multipli ... 319
Esempio 3: gruppi di tag multipli, tag singoli ... 321
Esempio 4: più gruppi di tag, più tag ... 323
Utilizzo delle istanze Amazon EC2 ... 327
Crea un'istanza Amazon EC2 per CodeDeploy ... 327
Crea un'istanza Amazon EC2 (modello)CloudFormation ... 334
Configurare un'istanza Amazon EC2 ... 345
Utilizzo delle istanze on-premises ... 349
Prerequisiti per la configurazione di un'istanza locale ... 350
Registra un'istanza on-premise ... 352
Gestione delle operazioni sulle istanze locali ... 389

Visualizzare i dettagli dell'istanza ... 396
Visualizza i dettagli dell'istanza (console) ... 396
Visualizza i dettagli dell'istanza (CLI) ... 397
Stato dell'istanza ... 398
Health status (Stato di integrità) ... 398
Informazioni sul numero minimo di istanze integre ... 400
Circa il numero minimo di istanze integre per zona di disponibilità ... 404
Utilizzo delle configurazioni della distribuzione ... 407
Configurazioni di distribuzione su una piattaforma di elaborazione EC2/on-premises ... 407
Configurazioni di implementazione predefinite ... 407
Configurazioni delle implementazioni su una piattaforma di calcolo Amazon ECS ... 412
Configurazioni di distribuzione predefinite per Amazon ECS ... 412
Configurazioni di distribuzione per CloudFormation blue/green distribuzioni (Amazon ECS) ... 413
Configurazioni di distribuzione su un AWS Lambda piattaforma di calcolo ... 414
Configurazioni di distribuzione predefinite per Lambda ... 414
Argomenti ... 415
Creare una configurazione di distribuzione ... 415
Creare una configurazione di distribuzione (console) ... 416
Creare una configurazione di distribuzione (AWS CLI) ... 419
Visualizza i dettagli della configurazione di distribuzione ... 420
Visualizza i dettagli della configurazione di distribuzione (console) ... 420
Visualizza la configurazione di distribuzione (CLI) ... 421
Eliminare una configurazione di distribuzione ... 421
Utilizzo delle applicazioni ... 423
Creazione di un'applicazione ... 424
Crea un'applicazione per un'implementazione sul posto (console) ... 426
Crea un'applicazione per una blue/green distribuzione (console) ... 429
Crea un'applicazione per l'implementazione di un servizio Amazon ECS (console) ... 434
Crea un'applicazione per un AWS Lambda implementazione della funzione (console) ... 436
Creare un'applicazione (CLI) ... 439
Visualizza i dettagli dell'applicazione ... 439
Visualizza i dettagli dell'applicazione (console) ... 439
Visualizza i dettagli dell'applicazione (CLI) ... 440
Creazione di una regola di notifica ... 440
Rinominare un'applicazione ... 443
Eliminazione di un'applicazione ... 443

Eliminare un'applicazione (console) 444
Eliminare un'applicazione (AWS CLI) 444
Utilizzo dei gruppi di distribuzione 445
Gruppi di istanze implementate nelle implementazioni della piattaforma di calcolo Amazon ECS 445
Gruppi di distribuzione in AWS Lambda implementazioni di piattaforme di calcolo 445
Gruppi di distribuzione nelle EC2/On-Premises implementazioni di Compute Platform 445
Argomenti 446
Creare un gruppo di distribuzione 447
Crea un gruppo di distribuzione per una distribuzione sul posto (console) 447
Crea un gruppo di distribuzione per una distribuzione (console) EC2/On-Premises blue/green 451
Crea un gruppo di distribuzione per una distribuzione Amazon ECS (console) 458
Configura un sistema di bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2 460
Configura un sistema di bilanciamento del carico, gruppi target e listener per le distribuzioni di CodeDeploy Amazon ECS 461
Creare un gruppo di distribuzione (CLI) 466
Visualizza i dettagli del gruppo di distribuzione 467
Visualizza i dettagli del gruppo di distribuzione (console) 468
Visualizza i dettagli del gruppo di distribuzione (CLI) 468
Modificare le impostazioni del gruppo di distribuzione 469
Modifica le impostazioni del gruppo di distribuzione (console) 469
Modifica delle impostazioni del gruppo di distribuzione (CLI) 470
Configurazione delle opzioni avanzate per un gruppo di distribuzione 472
Eliminare un gruppo di distribuzione 475
Eliminare un gruppo di distribuzione (console) 475
Eliminare un gruppo di distribuzione (CLI) 476
Utilizzo delle versioni delle applicazioni 477
Pianifica una revisione 477
Aggiungere un file AppSpec 478
Aggiungi un AppSpec file per una distribuzione Amazon ECS 478
Aggiungi un AppSpec file per un AWS Implementazione Lambda 481
Aggiungi un AppSpec file per una distribuzione EC2/On-Premises 483
Scegli un tipo di repository 487
Invia una revisione 490

Invia una revisione utilizzando il AWS CLI ... 492
Visualizza i dettagli della revisione dell'applicazione ... 494
Visualizza i dettagli sulla revisione dell'applicazione (console) ... 495
Visualizza i dettagli della revisione dell'applicazione (CLI) ... 495
Registra una revisione dell'applicazione ... 497
Registra una revisione in Amazon S3 con CodeDeploy (CLI) ... 497
Registrare una revisione GitHub con CodeDeploy (CLI) ... 498
Utilizzo delle distribuzioni ... 500
Crea distribuzione ... 501
Prerequisiti di distribuzione ... 502
Crea un'implementazione di Amazon ECS Compute Platform (console) ... 505
Crea un AWS Implementazione della piattaforma di elaborazione Lambda (console) ... 507
Crea una distribuzione di EC2/On-Premises Compute Platform (console) ... 509
Crea un'implementazione di Amazon ECS Compute Platform (CLI) ... 514
Crea un AWS Implementazione della piattaforma di elaborazione Lambda (CLI) ... 515
Crea un'implementazione di EC2/On-Premises Compute Platform (CLI) ... 517
Crea una blue/green distribuzione Amazon ECS tramite CloudFormation ... 521
Visualizza i dettagli della distribuzione ... 524
Visualizza i dettagli di distribuzione (console) ... 525
Visualizza i dettagli della distribuzione (CLI) ... 526
Visualizza i dati del registro di distribuzione ... 526
Visualizza i dati dei file di registro nella CloudWatch console Amazon ... 527
Visualizza i file di registro su un'istanza ... 527
Interrompere una distribuzione ... 531
Arresto di una distribuzione (console) ... 531
Arresto di una distribuzione (CLI) ... 532
Ridistribuire e ripristinare una distribuzione ... 532
Rollback automatici ... 533
Rollback manuali ... 533
Flusso di lavoro di rollback e ridistribuzione ... 534
Comportamento di rollback con il contenuto esistente ... 535
Distribuisci un'applicazione in un account diverso AWS ... 539
Passaggio 1: crea un bucket S3 in entrambi gli account ... 539
Fase 2: concedere le autorizzazioni del bucket Amazon S3 al profilo dell'istanza IAM dell'account di produzione ... 540
Fase 3: Creare risorse e un ruolo per più account nell'account di produzione ... 541

Fase 4: caricare la revisione dell'applicazione nel bucket Amazon S3 543
Fase 5: Assumi il ruolo di più account e distribuisci le applicazioni 543
Convalida un pacchetto di distribuzione su un computer locale 543
Prerequisiti 544
Crea una distribuzione locale 547
Esempi 550
Monitoraggio delle distribuzioni 552
Strumenti automatici 552
Strumenti manuali 554
Monitoraggio delle implementazioni con gli strumenti di Amazon CloudWatch 555
Monitoraggio delle implementazioni con allarmi CloudWatch 555
Monitoraggio delle distribuzioni con Amazon Events CloudWatch 557
Monitoraggio delle implementazioni con AWS CloudTrail 560
CodeDeploy informazioni in CloudTrail 560
Comprendere CodeDeploy le voci dei file di registro 561
Monitoraggio delle distribuzioni con le notifiche degli eventi di Amazon SNS 562
Concedi le autorizzazioni Amazon SNS per un ruolo di servizio 564
Crea un trigger per un CodeDeploy evento 565
Modifica un trigger in un gruppo di distribuzione 572
Eliminare un trigger da un gruppo di distribuzione 574
Formati di dati JSON per i trigger 575
Sicurezza 578
Protezione dei dati 578
Riservatezza del traffico Internet 580
Crittografia dei dati a riposo 580
Crittografia dei dati in transito 580
Gestione delle chiavi crittografiche 581
Gestione dell'identità e degli accessi 581
Destinatari 581
Autenticazione con identità 582
Gestione dell'accesso tramite policy 583
In che modo AWS CodeDeploy funziona con IAM 584
AWS politiche gestite (predefinite) per CodeDeploy 589
CodeDeploy aggiornamenti a AWS policy gestite 598
Identity-based esempi di politiche 602
Risoluzione dei problemi 609

CodeDeploy riferimento alle autorizzazioni ... 611
Cross-service confusa prevenzione sostitutiva ... 620
Risposta agli incidenti ... 622
Controllo di tutte le interazioni con CodeDeploy ... 622
Gestione degli avvisi e degli incidenti ... 622
Convalida della conformità ... 623
Resilienza ... 624
Sicurezza dell'infrastruttura ... 624
Documentazione di riferimento ... 625
AppSpec riferimento al file ... 625
AppSpec file su una piattaforma di elaborazione Amazon ECS ... 626
AppSpec file su un AWS Lambda piattaforma di calcolo ... 626
AppSpec file su una piattaforma di EC2/on-premises elaborazione ... 626
AppSpec Struttura dei file ... 627
AppSpec Esempio di file ... 673
AppSpec Spaziatura tra i file ... 679
Convalida il AppSpec file e la posizione del file ... 681
Riferimento alla configurazione dell'agente ... 681
Argomenti correlati ... 694
CloudFormation riferimento al modello ... 694
Utilizzo CodeDeploy con Amazon Virtual Private Cloud ... 697
Disponibilità ... 698
Crea endpoint VPC per CodeDeploy ... 701
Configura l'agente e le autorizzazioni IAM CodeDeploy ... 702
Riferimento al kit di risorse ... 703
Nomi dei bucket del Resource Kit per regione ... 703
Contenuto del kit di risorse ... 706
Visualizza un elenco dei file del Resource Kit ... 708
Scarica i file del kit di risorse ... 709
Quote ... 713
Risoluzione dei problemi ... 720
Risoluzione di problemi generali ... 720
Lista di controllo per la risoluzione di problemi generali ... 721
CodeDeploy le risorse di distribuzione sono supportate solo in alcune AWS Regioni ... 722
Le procedure descritte in questa guida non corrispondono a quelle della CodeDeploy console ... 723

I ruoli IAM richiesti non sono disponibili 723
L'utilizzo di alcuni editor di testo per creare AppSpec file e script di shell può causare il fallimento delle distribuzioni 723
L'utilizzo di Finder in macOS per includere in un file compresso la revisione di un'applicazione può determinare l'esito negativo delle distribuzioni 724
Risolvi i problemi EC2/On-Premises di distribuzione 725
CodeDeploy errore relativo alle credenziali mancanti del plugin CommandPoller 726
La distribuzione non riesce con il messaggio «Convalida del messaggio firmato PKCS7 fallita» 726
La distribuzione o la ridistribuzione degli stessi file alle stesse posizioni delle istanze avrà esito negativo e mostrerà l'errore "La distribuzione non è riuscita a causa di un file specificato già esistente in questa posizione" 726
I percorsi di file lunghi causano errori del tipo «Nessun file o directory di questo tipo» 729
Long-running i processi possono causare il fallimento delle distribuzioni 730
Risoluzione dei problemi relativi a un evento AllowTraffic del ciclo di vita non riuscito senza che sia stato riportato alcun errore nei log di distribuzione 732
Risoluzione dei problemi relativi a un evento non riuscito o ApplicationStop AfterBlockTraffic relativo al BeforeBlockTraffic ciclo di vita dell'implementazione 732
Risoluzione dei problemi relativi a un evento del ciclo di vita di DownloadBundle distribuzione non riuscito con UnknownError: not open-for reading 734
Risoluzione dei problemi relativi a tutti gli eventi del ciclo di vita ignorati 735
Per impostazione predefinita, PowerShell gli script di Windows non riescono a utilizzare la versione a 64 bit di Windows PowerShell 736
Risolvi i problemi di distribuzione di Amazon ECS 738
Si verifica un timeout durante l'attesa del set di attività sostitutive 738
Si verifica un timeout in attesa del proseguimento di una notifica 739
Il ruolo IAM non dispone di autorizzazioni sufficienti 740
La distribuzione è scaduta in attesa di una richiamata di stato 740
La distribuzione non è riuscita perché una o più delle funzioni di convalida degli eventi del ciclo di vita non sono riuscite 741
L'ELB non può essere aggiornato a causa del seguente errore: Il gruppo target del taskset primario deve essere dietro l'ascoltatore 741
La mia distribuzione a volte non riesce quando utilizzo Auto Scaling 742
Solo ALB supporta il routing graduale del traffico, utilizza invece il routing AllAtOnce del traffico quando crei un gruppo di distribuzione create/update 743

Anche se la mia distribuzione è riuscita, il set di attività sostitutivo non supera i controlli di integrità di Elastic Load Balancing e la mia applicazione è inattiva 744
Posso collegare più sistemi di bilanciamento del carico a un gruppo di distribuzione? 745
Posso eseguire CodeDeploy blue/green implementazioni senza un sistema di bilanciamento del carico? 745
Come posso aggiornare il mio servizio Amazon ECS con nuove informazioni durante una distribuzione? 745
Risoluzione dei problemi AWS Problemi di implementazione Lambda 746
AWS Lambda le distribuzioni falliscono dopo l'interruzione manuale di una distribuzione Lambda che non dispone di rollback configurati 746
Risoluzione dei problemi relativi ai gruppi di distribuzione 747
Il tagging di un'istanza come parte di un gruppo di distribuzione non comporta automaticamente la distribuzione dell'applicazione nella nuova istanza 747
Risoluzione dei problemi relativi alle istanze 747
I tag devono essere impostati correttamente 747
AWS CodeDeploy l'agente deve essere installato e in esecuzione sulle istanze 748
Quando un'istanza viene terminata durante una distribuzione, la distribuzione continua senza errori per un'ora 748
Analisi dei file di log per individuare gli errori di distribuzione nelle istanze 748
Crea un nuovo file di CodeDeploy registro se è stato eliminato accidentalmente 749
Risoluzione dei problemi relativi alla distribuzione «InvalidSignatureException — Firma scaduta: [ora] è precedente a [ora]» 749
Risolvi i problemi relativi GitHub ai token 750
Token OAuth non valido GitHub 750
È stato superato GitHub il numero massimo di token OAuth 750
Risolvi i problemi relativi ad Amazon EC2 Auto Scaling 751
Risoluzione dei problemi generali di Amazon EC2 Auto Scaling 751
"CodeDeployRole non consente di eseguire le seguenti operazioni AWS service: AmazonAutoScaling "errore 752
Le istanze in un gruppo Amazon EC2 Auto Scaling vengono continuamente fornite e terminate prima di poter implementare una revisione 753
L'interruzione o il riavvio di un'istanza di Amazon EC2 Auto Scaling potrebbe causare il fallimento delle distribuzioni 754
Evita di associare più gruppi di distribuzione a un singolo gruppo Amazon EC2 Auto Scaling 754

Le istanze EC2 in un gruppo Amazon EC2 Auto Scaling non vengono avviate e ricevono l'errore «Heartbeat Timeout» 755
I collegamenti non corrispondenti al ciclo di vita di Amazon EC2 Auto Scaling potrebbero causare l'interruzione o il fallimento delle distribuzioni automatiche nei gruppi di Amazon EC2 Auto Scaling 758
Errore «La distribuzione non è riuscita perché non è stata trovata alcuna istanza per il tuo gruppo di distribuzione» 759
Codici di errore 767
Argomenti correlati 773
Resources 774
Guide di riferimento e risorse di supporto 774
Esempi 774
Blog 774
AWS kit e strumenti di sviluppo software 774
Cronologia dei documenti 776
Aggiornamenti precedenti 796
AWS Glossario 821
..... dcccxxii

Che cos'è CodeDeploy?

CodeDeploy è un servizio di distribuzione che automatizza le distribuzioni di applicazioni su istanze Amazon EC2, istanze locali, funzioni Lambda serverless o servizi Amazon ECS.

È possibile distribuire una varietà quasi illimitata di contenuti di applicazioni, tra cui:

- Codice
- Funzioni serverless AWS Lambda
- File Web e di configurazione
- Eseguibili
- Pacchetti
- Script
- File multimediali

CodeDeploy può distribuire contenuti applicativi eseguiti su un server e archiviati in bucket GitHub , repository o repository Bitbucket di Amazon S3. CodeDeploy può anche implementare una funzione Lambda serverless. Non è necessario modificare il codice esistente prima di poter utilizzare CodeDeploy.

CodeDeploy ti consente di:

- Il rilascio rapido di nuove funzionalità.
- Aggiorna le versioni delle AWS Lambda funzioni.
- Evitare tempi di inattività durante la distribuzione dell'applicazione.
- La gestione della complessità dell'aggiornamento delle applicazioni, senza molti dei rischi associati alle distribuzioni manuali soggette a errori.

Il servizio si dimensiona con l'infrastruttura in modo da permetterti di distribuire facilmente a una sola o a migliaia di istanze.

CodeDeploy funziona con vari sistemi per la gestione della configurazione, il controllo del codice sorgente, l'integrazione continua, la distribuzione continua e l'implementazione continua. Per ulteriori informazioni, consulta Integrazioni di prodotti.

La CodeDeploy console offre anche un modo per cercare rapidamente le risorse, come repository, progetti di creazione, applicazioni di distribuzione e pipeline. Scegli Go to resource (Vai alla risorsa) o premi il tasto / e digita il nome della risorsa. Qualsiasi corrispondenza verrà visualizzata nell'elenco. Le ricerche rispettano la distinzione tra maiuscole e minuscole. Puoi visualizzare solo le risorse per le quali disponi dell'autorizzazione di visualizzazione. Per ulteriori informazioni, consulta Gestione delle identità e degli accessi per AWS CodeDeploy.

Argomenti

- Vantaggi di AWS CodeDeploy
- Panoramica delle piattaforme di elaborazione CodeDeploy
- Panoramica dei tipi di implementazione CodeDeploy
- Ci piacerebbe conoscere la tua opinione
- CodeDeploy componenti primari
- CodeDeploy implementazioni
- CodeDeploy file delle specifiche dell'applicazione (AppSpec)

Vantaggi di AWS CodeDeploy

CodeDeploy offre i seguenti vantaggi:

- Applicazioni server, serverless e container. CodeDeployconsente di distribuire sia applicazioni tradizionali su server sia applicazioni che distribuiscono una versione con AWS Lambda funzioni serverless o un'applicazione Amazon ECS.
- Implementazioni automatizzate. CodeDeploy automatizza completamente le implementazioni delle applicazioni negli ambienti di sviluppo, test e produzione. CodeDeploysi adatta alla tua infrastruttura in modo da poter eseguire l'implementazione su una o migliaia di istanze.
- Ridurre al minimo i tempi di inattività. Se la tua applicazione utilizza la piattaforma di EC2/On-Premises elaborazione, CodeDeploy aiuta a massimizzare la disponibilità dell'applicazione. Durante una distribuzione sul posto, CodeDeploy esegue un aggiornamento continuo tra le istanze Amazon EC2. È possibile specificare il numero di istanze da mettere offline alla volta per gli aggiornamenti. Durante una blue/green distribuzione, la revisione più recente dell'applicazione viene installata sulle istanze sostitutive. Il traffico è reindirizzato a queste istanze quando si sceglie, immediatamente o non appena terminato il test, il nuovo ambiente. Per entrambi i tipi di distribuzione, CodeDeploy monitora lo stato delle applicazioni in base alle regole configurate.

- Interrompere ed eseguire il rollback. È possibile arrestare ed eseguire il rollback automaticamente o manualmente sulle distribuzioni in caso di errore.
- Controllo centralizzato. È possibile avviare e monitorare lo stato delle distribuzioni tramite la console o il CodeDeploy . AWS CLI Riceverai un report che elenca quando è stata distribuita ogni revisione dell'applicazione e su quali istanze Amazon EC2.
- Facile da adottare. CodeDeploy è indipendente dalla piattaforma e funziona con qualsiasi applicazione. Puoi riutilizzare facilmente il codice di configurazione. CodeDeploy può anche integrarsi con il processo di rilascio del software o la toolchain di distribuzione continua.
- Distribuzioni simultanee Se disponi di più di un'applicazione che utilizza la piattaforma di EC2/On-Premises elaborazione, CodeDeploy puoi distribuirle contemporaneamente sullo stesso set di istanze.

Panoramica delle piattaforme di elaborazione CodeDeploy

CodeDeploy è in grado di distribuire applicazioni su tre piattaforme di elaborazione:

- EC2/On-Premises: descrive le istanze di server fisici che possono essere istanze cloud Amazon EC2, server locali o entrambi. Le applicazioni create utilizzando la piattaforma di EC2/On-Premises calcolo possono essere composte da file eseguibili, file di configurazione, immagini e altro ancora.

  Le distribuzioni che utilizzano la piattaforma di EC2/On-Premises elaborazione gestiscono il modo in cui il traffico viene indirizzato alle istanze utilizzando un tipo di distribuzione locale o locale. blue/green Per ulteriori informazioni, consulta Panoramica dei tipi di implementazione CodeDeploy.
- AWS Lambda: utilizzato per distribuire applicazioni costituite da una versione aggiornata di una funzione Lambda. AWS Lambda gestisce la funzione Lambda in un ambiente di calcolo serverless costituito da una struttura di calcolo ad alta disponibilità. Tutta l'amministrazione delle risorse di calcolo viene eseguita da. AWS Lambda Per ulteriori informazioni, vedere Serverless Computing and Applications. Per ulteriori informazioni sulle AWS Lambda funzioni Lambda, vedere. AWS Lambda

  Puoi gestire il modo in cui il traffico viene spostato verso le versioni aggiornate della funzione Lambda durante un'implementazione scegliendo una configurazione canaria, lineare o completa.
- Amazon ECS: utilizzato per distribuire un'applicazione containerizzata Amazon ECS come set di attività. CodeDeploy esegue una blue/green distribuzione installando una versione aggiornata dell'applicazione come nuovo set di attività sostitutivo. CodeDeploy reindirizza il traffico di

produzione dal set di attività dell'applicazione originale al set di attività sostitutivo. Il set di attività originale viene terminato una volta completata l'implementazione. Per ulteriori informazioni su Amazon ECS, consulta Amazon Elastic Container Service.

È possibile gestire il modo in cui il traffico viene spostato al set di attività aggiornato durante una distribuzione scegliendo una configurazione canary, lineare o all-at-once.

> **Note**
>
> Le blue/green implementazioni di Amazon ECS sono supportate utilizzando entrambi e. CodeDeploy CloudFormation I dettagli relativi a queste distribuzioni sono descritti nelle sezioni successive.

La tabella seguente descrive come CodeDeploy i componenti vengono utilizzati con ciascuna piattaforma di elaborazione. Per ulteriori informazioni, consulta:

- Lavorare con i gruppi di distribuzione in CodeDeploy
- Utilizzo delle distribuzioni in CodeDeploy
- Utilizzo delle configurazioni di distribuzione in CodeDeploy
- Utilizzo delle revisioni delle applicazioni per CodeDeploy
- Utilizzo delle applicazioni in CodeDeploy

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
Gruppo di distribuzione	Distribuisce una revisione a un set di istanze.	Implementa una nuova versione di una funzione Lambda serverless su un'infrastruttura di elaborazione ad alta disponibilità.	Speciifica il servizio Amazon ECS con l'applica zione container izzata da distribuire come set di

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
			attività, un listener di produzione e e di test opzionale utilizzato per servire il traffico verso l'applicazione distribuita, quando reindirizzare il traffico e terminare il set di attività originale dell'applicazione distribuita e le impostazioni opzionali di attivazione, allarme e rollback.

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
Implementazione	Implementa una nuova revisione composta da un'applicazione e un file. AppSpec AppSpec specifica come distribuire l'applicazione sulle istanze di un gruppo di distribuzione.	Sposta il traffico di produzione da una versione di una funzione Lambda a una nuova versione della stessa funzione. Il AppSpec file specifica quale versione della funzione Lambda distribuire.	Distribuisce una versione aggiornata di un'applicazione containerizzata Amazon ECS come nuovo set di attività sostitutivo. CodeDeploy reindirizza il traffico di produzione dal set di attività con la versione originale al nuovo set di attività sostitutivo con la versione aggiornata. Quando la distribuzione giunge a completam

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
			ento, il set di attività originale viene terminato.
Configurazione dell'implementazione	Impostazioni che determinano la velocità di distribuzione e il numero minimo di istanze che devono essere integre in qualsiasi momento durante una distribuzione.	Impostazioni che determinano in che modo il traffico viene spostato verso le versioni aggiornate della funzione Lambda.	Impostazioni che determinano in che modo il traffico viene spostato verso il set di attività Amazon ECS aggiornato.

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
Revision	Una combinazione di AppSpec file e file di applicazione, come file eseguibili, file di configurazione e così via.	Un AppSpec file che specifica la funzione Lambda da distribuire e le funzioni Lambda che possono eseguire test di convalida durante gli hook degli eventi del ciclo di vita della distribuzione.	Un AppSpec file che specifica: • La definizione delle attività Amazon ECS per il servizio Amazon ECS con l'applicazione containerizzata da distribuire. • Il container in cui l'applicazione aggiornata viene distribuita. • Una porta per il container

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
			in cui il traffico di produzione viene reindirizzato. • Impostazioni di configurazione di rete opzionali e funzioni Lambda che possono eseguire test di convalida durante gli hook degli eventi del ciclo di vita dell'implementazione.

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
Applicazione	Raccolta di gruppi di distribuzione e revisioni. Un' EC2/On-Premisesapplicazione utilizza la piattaforma di elaborazione. EC2/On-Premises	Raccolta di gruppi di distribuzione e revisioni. Un'applicazione utilizzata per una AWS Lambda distribuzione utilizza la piattaforma di calcolo AWS Lambda serverless.	Raccolta di gruppi di distribuzione e revisioni. Un'applicazione utilizzata per una distribuzione Amazon ECS utilizza la piattaforma di calcolo Amazon ECS.

Panoramica dei tipi di implementazione CodeDeploy

CodeDeploy offre due opzioni di tipo di distribuzione:

- In-place distribuzione: l'applicazione su ogni istanza del gruppo di distribuzione viene interrotta, viene installata la revisione più recente dell'applicazione e la nuova versione dell'applicazione viene avviata e convalidata. È possibile utilizzare un sistema di bilanciamento del carico in modo che ogni istanza venga annullata durante la distribuzione e quindi ripristinata in servizio al termine della distribuzione. Solo le distribuzioni che utilizzano la piattaforma di EC2/On-Premises elaborazione possono utilizzare le distribuzioni sul posto. Per ulteriori informazioni sulle implementazioni sul posto, consulta. Panoramica di una distribuzione sul posto

> **Note**
>
> AWS Le distribuzioni Lambda e Amazon ECS non possono utilizzare un tipo di distribuzione sul posto.

- Blue/green implementazione: il comportamento della distribuzione dipende dalla piattaforma di elaborazione utilizzata:
  - Blue/green su una piattaforma di EC2/On-Premises elaborazione: le istanze in un gruppo di distribuzione (l'ambiente originale) vengono sostituite da un diverso set di istanze (l'ambiente sostitutivo) utilizzando questi passaggi:
    - Le istanze vengono fornite per l'ambiente sostitutivo.
    - La revisione più recente dell'applicazione viene installata sulle istanze sostitutive.
    - Si verifica un tempo di attesa opzionale per attività come il test delle applicazioni e la verifica del sistema.
    - Le istanze nell'ambiente sostitutivo vengono registrate con uno o più sistemi di bilanciamento del carico Elastic Load Balancing, causando il reindirizzamento del traffico verso di essi. Le istanze nell'ambiente originale vengono cancellate e possono essere terminate o mantenute in esecuzione per altri usi.

    > **Note**
    >
    > Se utilizzi una piattaforma di EC2/On-Premises elaborazione, tieni presente che le blue/green distribuzioni funzionano solo con istanze Amazon EC2.

  - Blue/green su una piattaforma AWS Lambda di calcolo Amazon ECS: il traffico viene spostato in incrementi in base a una configurazione di distribuzione canaria, lineare o completa.
  - Blue/green implementazioni tramite CloudFormation: Il traffico viene spostato dalle risorse correnti alle risorse aggiornate come parte di un aggiornamento dello stack. CloudFormation Attualmente sono supportate solo le implementazioni ECS blue/green .

Per ulteriori informazioni sulle blue/green distribuzioni, consulta. Panoramica di una distribuzione blue/green

> **Note**
>
> Utilizzando l' CodeDeploy agente, è possibile eseguire una distribuzione su un'istanza a cui si è effettuato l'accesso senza la necessità di un'applicazione, di un gruppo di distribuzione o persino di un AWS account. Per informazioni, consulta Utilizzare l' CodeDeploy agente per convalidare un pacchetto di distribuzione su un computer locale.

Argomenti

- Panoramica di una distribuzione sul posto
- Panoramica di una distribuzione blue/green

Panoramica di una distribuzione sul posto

> **Note**
>
> AWS Le distribuzioni Lambda e Amazon ECS non possono utilizzare un tipo di distribuzione sul posto.

Ecco come funziona una distribuzione sul posto:

1. Innanzitutto, si creano contenuti distribuibili sulla macchina di sviluppo locale o in un ambiente simile, quindi si aggiunge un file (file) AppSpec con le specifiche dell'applicazione. Il AppSpec file è unico per. CodeDeploy Definisce le azioni di distribuzione che si CodeDeploy desidera eseguire. Raggruppi il contenuto distribuibile e il AppSpec file in un file di archivio, quindi lo carichi in un bucket Amazon S3 o in un repository. GitHub Questo file di archivio viene chiamato revisione dell'applicazione (o semplicemente revisione).
2. Successivamente, fornisci CodeDeploy informazioni sulla tua distribuzione, ad esempio da quale bucket o GitHub repository Amazon S3 estrarre la revisione e su quale set di istanze Amazon EC2 distribuirne il contenuto. CodeDeploy chiama un insieme di istanze Amazon EC2 un gruppo di distribuzione. Un gruppo di distribuzione contiene istanze Amazon EC2 con tag individuali, istanze Amazon EC2 in gruppi Amazon EC2 Auto Scaling o entrambi.

   Ogni volta che carichi con successo una nuova revisione di un'applicazione che desideri distribuire al gruppo di distribuzione, quel bundle viene impostato come revisione di destinazione per il gruppo di distribuzione. In altre parole, la revisione dell'applicazione attualmente destinata alla

distribuzione è la revisione target, che è anche la revisione che viene utilizzata per le distribuzioni automatiche.

3. Successivamente, l' CodeDeploy agente di ogni istanza esegue un sondaggio CodeDeploy per determinare cosa e quando estrarre dal GitHub bucket o dal repository Amazon S3 specificato.
4. Infine, l' CodeDeploy agente di ogni istanza estrae la revisione di destinazione dal bucket GitHub o dal repository Amazon S3 e, utilizzando le istruzioni nel file, distribuisce AppSpec i contenuti nell'istanza.

CodeDeploy tiene traccia delle distribuzioni in modo da poter conoscere lo stato della distribuzione, i parametri di configurazione della distribuzione, lo stato dell'istanza e così via.

Panoramica di una distribuzione blue/green

Una blue/green distribuzione viene utilizzata per aggiornare le applicazioni riducendo al minimo le interruzioni causate dalle modifiche di una nuova versione dell'applicazione. CodeDeploy effettua il provisioning della nuova versione dell'applicazione insieme a quella precedente prima di reindirizzare il traffico di produzione.

- AWS Lambda: il traffico viene spostato da una versione di una funzione Lambda a una nuova versione della stessa funzione Lambda.
- Amazon ECS: il traffico viene spostato da un'attività impostata nel servizio Amazon ECS a un'attività aggiornata e sostitutiva impostata nello stesso servizio Amazon ECS.
- EC2/On-Premises: Il traffico viene reindirizzato da un set di istanze nell'ambiente originale a un set di istanze di sostituzione.

Tutte le AWS implementazioni Lambda e Amazon ECS lo sono. blue/green Una EC2/On-Premises distribuzione può avvenire sul posto o. blue/green Un' blue/green implementazione offre una serie di vantaggi rispetto a un'implementazione sul posto:

- È possibile installare e testare un'applicazione nel nuovo ambiente di sostituzione e distribuirla alla produzione semplicemente reindirizzando il traffico.
- Se utilizzi la piattaforma di EC2/On-Premises elaborazione, tornare alla versione più recente di un'applicazione è più veloce e affidabile. Questo perché il traffico può essere instradato nuovamente verso le istanze originali a patto che non siano state terminate. Con una distribuzione in loco, è necessario eseguire il rollback delle versioni ridistribuendo la versione precedente dell'applicazione.

- Se utilizzi la piattaforma di EC2/On-Premises elaborazione, vengono predisposte nuove istanze per una blue/green distribuzione e riflettono le configurazioni del server più aggiornate. In questo modo, è possibile evitare i tipi di problemi talvolta associati alle istanze con esecuzione prolungata.
- Se utilizzi la piattaforma di calcolo AWS Lambda, puoi controllare come il traffico viene spostato dalla versione originale della funzione Lambda alla nuova versione della AWS funzione Lambda. AWS
- Se utilizzi la piattaforma di calcolo Amazon ECS, puoi controllare come il traffico viene spostato dal set di attività originale al nuovo set di attività.

Una blue/green distribuzione con CloudFormation può utilizzare uno dei seguenti metodi:

- CloudFormation modelli per le distribuzioni: quando si configurano le distribuzioni con i CloudFormation modelli, le distribuzioni vengono attivate dagli aggiornamenti. CloudFormation Quando modifichi una risorsa e carichi una modifica al modello, un aggiornamento dello stack avvia la nuova distribuzione. CloudFormation Per un elenco delle risorse che puoi utilizzare nei CloudFormation modelli, consulta. CloudFormation modelli di CodeDeploy riferimento
- Blue/green distribuzioni tramite CloudFormation: È possibile utilizzare CloudFormation per gestire le blue/green distribuzioni tramite aggiornamenti dello stack. È possibile definire le risorse blu e verdi, oltre a specificare le impostazioni di routing e stabilizzazione del traffico, all'interno del modello di stack. Quindi, se aggiorni risorse selezionate durante un aggiornamento dello stack, CloudFormation genera tutte le risorse verdi necessarie, sposta il traffico in base ai parametri di routing del traffico specificati ed elimina le risorse blu. Per ulteriori informazioni, consulta Automatizza le blue/green distribuzioni di Amazon ECS tramite l' CodeDeploy utilizzo CloudFormation nella Guida per l'utente.AWS CloudFormation

  > **Note**
  >
  > Supportato solo per le blue/green distribuzioni Amazon ECS.

Il modo in cui configuri una blue/green distribuzione dipende dalla piattaforma di elaborazione utilizzata dalla distribuzione.

Blue/Green distribuzione su un AWS Lambda o piattaforma di elaborazione Amazon ECS

Se utilizzi la piattaforma di calcolo Amazon ECS AWS Lambda o Amazon ECS, devi indicare in che modo il traffico viene spostato dalla funzione originale AWS Lambda o dal set di attività Amazon ECS alla nuova funzione o set di attività. Per indicare come viene spostato il traffico, devi specificare una delle seguenti configurazioni di distribuzione:

- canarino
- lineare
- tutto in una volta

Per informazioni su come viene spostato il traffico in configurazioni di distribuzione canarie, lineari o complete, consulta. Configurazione dell'implementazione

Per i dettagli sulla configurazione della distribuzione Lambda, consulta. Configurazioni di distribuzione su un AWS Lambda piattaforma di calcolo

Per dettagli sulla configurazione di distribuzione di Amazon ECS, consultaConfigurazioni delle implementazioni su una piattaforma di calcolo Amazon ECS.

Blue/Green implementazione su una piattaforma di EC2/on-premises elaborazione

> **Note**
>
> È necessario utilizzare istanze Amazon EC2 per le blue/green distribuzioni sulla piattaforma di elaborazione. EC2/On-Premises On-premises le istanze non sono supportate per il tipo di implementazione. blue/green

Se utilizzi la piattaforma di EC2/On-Premises elaborazione, vale quanto segue:

È necessario disporre di una o più istanze Amazon EC2 con tag di identificazione Amazon EC2 o di un gruppo Amazon EC2 Auto Scaling. Le istanze devono soddisfare anche i seguenti requisiti:

- Ogni istanza Amazon EC2 deve avere il profilo di istanza IAM corretto collegato.
- L' CodeDeploy agente deve essere installato e in esecuzione su ogni istanza.

> **Note**
>
> In genere è inoltre in esecuzione una revisione dell'applicazione sulle istanze dell'ambiente originale, ma questo non è un requisito per la distribuzione. blue/green

Quando si crea un gruppo di distribuzione utilizzato nelle blue/green distribuzioni, è possibile scegliere come specificare l'ambiente sostitutivo:

Copia un gruppo Amazon EC2 Auto Scaling esistente: blue/green durante la CodeDeploy distribuzione, crea le istanze per il tuo ambiente sostitutivo durante la distribuzione. Con questa opzione, CodeDeploy utilizza il gruppo Amazon EC2 Auto Scaling specificato come modello per l'ambiente sostitutivo, incluso lo stesso numero di istanze in esecuzione e molte altre opzioni di configurazione.

Scegli le istanze manualmente: puoi specificare le istanze da conteggiare come sostitutive utilizzando i tag delle istanze Amazon EC2, i nomi dei gruppi di Amazon EC2 Auto Scaling o entrambi. Se scegli questa opzione, non devi specificare le istanze per l'ambiente di sostituzione fino a quando non crei una distribuzione.

Come funziona:

1. Hai già delle istanze o un gruppo Amazon EC2 Auto Scaling che funge da ambiente originale. La prima volta che esegui una blue/green distribuzione, in genere utilizzi istanze già utilizzate in una distribuzione locale.
2. In un' CodeDeploy applicazione esistente, si crea un gruppo di blue/green distribuzione in cui, oltre alle opzioni richieste per una distribuzione sul posto, si specifica quanto segue:
   - Il sistema di bilanciamento del carico o i sistemi di bilanciamento del carico che indirizzano il traffico dall'ambiente originale all'ambiente sostitutivo durante il processo di distribuzione. blue/green
   - Se reinstradare il traffico verso l'ambiente di sostituzione immediatamente o attendere di reinstradarlo manualmente.
   - La velocità in cui il traffico viene instradato verso le istanze di sostituzione.
   - Se le istanze sostituite vengono terminate o lasciate in esecuzione.
3. Crei una distribuzione per questo gruppo di distribuzione durante la quale si verifica quanto segue:
   a. Se scegli di copiare un gruppo Amazon EC2 Auto Scaling, le istanze vengono fornite per l'ambiente sostitutivo.

b. La revisione dell'applicazione che specifichi per la distribuzione viene installata sulle istanze di sostituzione.
c. Se è stato specificato un tempo di attesa nelle impostazioni del gruppo di distribuzione, la distribuzione viene messa in pausa. Questo è il momento in cui puoi eseguire i test e le verifiche nell'ambiente di sostituzione. Se non reinstradi manualmente il traffico prima della fine del periodo di attesa, la distribuzione viene arrestata.
d. Le istanze nell'ambiente sostitutivo vengono registrate con un sistema di bilanciamento del carico Elastic Load Balancing e il traffico inizia a essere indirizzato verso di esse.
e. Viene annullata la registrazione delle istanze nell'ambiente originale e, in base a ciò che hai specificato nel gruppo di distribuzione, tali istanze vengono terminate o lasciate in esecuzione.

Blue/Green distribuzione tramite CloudFormation

È possibile gestire le CodeDeploy blue/green distribuzioni modellando le risorse con un modello. CloudFormation

Quando si modellano le blue/green risorse utilizzando un CloudFormation modello, si crea un aggiornamento dello stack CloudFormation che aggiorna il set di attività. Il traffico di produzione passa dal set di attività originale del servizio a un set di attività sostitutivo tutto in una volta, con distribuzioni lineari e tempi di bake o con distribuzioni canary. L'aggiornamento dello stack avvia una distribuzione in CodeDeploy. È possibile visualizzare lo stato e la cronologia della distribuzione in CodeDeploy, ma non è possibile creare o gestire CodeDeploy risorse al di fuori del CloudFormation modello.

> **Note**
>
> Per le blue/green distribuzioni tramite CloudFormation, non è necessario creare un' CodeDeploy applicazione o un gruppo di distribuzione.

Questo metodo supporta solo le blue/green distribuzioni di Amazon ECS. Per ulteriori informazioni sulle blue/green distribuzioni tramite, consulta. CloudFormationCrea una blue/green distribuzione Amazon ECS tramite CloudFormation

Ci piacerebbe conoscere la tua opinione

Apprezziamo il tuo feedback. Per contattarci, visita il CodeDeploy forum.

Argomenti

- Primary Components
- Deployments
- Application Specification Files

CodeDeploy componenti primari

Prima di iniziare a utilizzare il servizio, è necessario acquisire familiarità con i componenti principali del processo di distribuzione. CodeDeploy

Argomenti

- Applicazione
- Compute platform (Piattaforma di calcolo)
- Configurazione dell'implementazione
- Gruppo di distribuzione
- Il tipo di implementazione
- Profilo dell'istanza IAM
- Revision
- Ruolo del servizio
- Revisione dell'obiettivo
- Altri componenti

Applicazione

Un'applicazione è un nome che identifica in modo univoco l'applicazione che desideri distribuire. CodeDeploy utilizza questo nome, che funge da contenitore, per garantire che durante una distribuzione venga fatto riferimento alla combinazione corretta di revisione, configurazione di distribuzione e gruppo di distribuzione.

Compute platform (Piattaforma di calcolo)

Una piattaforma di elaborazione è una piattaforma su cui CodeDeploy distribuisce un'applicazione. Esistono tre piattaforme di elaborazione:

- EC2/On-Premises: descrive le istanze di server fisici che possono essere istanze cloud di Amazon EC2, server locali o entrambi. Le applicazioni create utilizzando la piattaforma di EC2/On-Premises calcolo possono essere composte da file eseguibili, file di configurazione, immagini e altro ancora.

  Le distribuzioni che utilizzano la piattaforma di EC2/On-Premises elaborazione gestiscono il modo in cui il traffico viene indirizzato alle istanze utilizzando un tipo di distribuzione locale o locale. blue/green Per ulteriori informazioni, consulta Panoramica dei tipi di implementazione CodeDeploy.
- AWS Lambda: utilizzato per distribuire applicazioni costituite da una versione aggiornata di una funzione Lambda. AWS Lambda gestisce la funzione Lambda in un ambiente di calcolo serverless costituito da una struttura di calcolo ad alta disponibilità. Tutta l'amministrazione delle risorse di calcolo viene eseguita da. AWS Lambda Per ulteriori informazioni, vedere Serverless Computing and Applications. Per ulteriori informazioni sulle AWS Lambda funzioni Lambda, vedere. AWS Lambda

  Puoi gestire il modo in cui il traffico viene spostato verso le versioni aggiornate della funzione Lambda durante un'implementazione scegliendo una configurazione canaria, lineare o completa.
- Amazon ECS: utilizzato per distribuire un'applicazione containerizzata Amazon ECS come set di attività. CodeDeploy esegue una blue/green distribuzione installando una versione aggiornata dell'applicazione come nuovo set di attività sostitutivo. CodeDeploy reindirizza il traffico di produzione dal set di attività dell'applicazione originale al set di attività sostitutivo. Il set di attività originale viene terminato una volta completata l'implementazione. Per ulteriori informazioni su Amazon ECS, consulta Amazon Elastic Container Service.

  È possibile gestire il modo in cui il traffico viene spostato al set di attività aggiornato durante una distribuzione scegliendo una configurazione canary, lineare o all-at-once.

> **Note**
>
> Le blue/green implementazioni di Amazon ECS sono supportate sia tramite che. CodeDeploy CloudFormation I dettagli relativi a queste distribuzioni sono descritti nelle sezioni successive.

Configurazione dell'implementazione

Una configurazione di distribuzione è un insieme di regole di distribuzione e condizioni di successo e fallimento della distribuzione utilizzate CodeDeploy durante una distribuzione. Se la distribuzione utilizza la piattaforma di EC2/On-Premises elaborazione, è possibile specificare il numero minimo di

istanze integre per la distribuzione. Se la tua distribuzione utilizza la piattaforma di calcolo Amazon ECS AWS Lambda o la piattaforma di calcolo Amazon ECS, puoi specificare in che modo il traffico viene indirizzato alla funzione Lambda aggiornata o al set di attività ECS.

Per ulteriori informazioni su come specificare il numero minimo di host integri per una distribuzione che utilizza la piattaforma di calcolo, consulta. EC2/On-Premises Informazioni sul numero minimo di istanze integre

Le seguenti configurazioni di distribuzione specificano come viene instradato il traffico durante una distribuzione che utilizza la piattaforma di calcolo Lambda o ECS:

- Canary: il traffico viene trasferito in due incrementi. Puoi scegliere tra opzioni canarie predefinite che specificano la percentuale di traffico trasferito alla funzione Lambda aggiornata o all'attività ECS impostata nel primo incremento e l'intervallo, in minuti, prima che il traffico rimanente venga spostato nel secondo incremento.
- Lineare: il traffico viene trasferito in incrementi uguali con lo stesso intervallo di tempo, in minuti, tra ciascun incremento. Puoi scegliere tra opzioni lineari predefinite che specificano la percentuale del traffico trasferito in ogni incremento e l'intervallo di tempo, in minuti, tra ciascun incremento.
- All-at-once: tutto il traffico viene spostato contemporaneamente dalla funzione Lambda o dal set di attività ECS originale alla funzione o al set di attività aggiornato.

Gruppo di distribuzione

Un gruppo di distribuzione è un insieme di singole istanze. Un gruppo di distribuzione contiene istanze con tag individuali, istanze Amazon EC2 in gruppi Amazon EC2 Auto Scaling o entrambi. Per informazioni sui tag delle istanze Amazon EC2, consulta Working with Tags Using the Console. Per informazioni sulle istanze locali, consulta Working with On-Premises Instances. Per informazioni su Amazon EC2 Auto Integrazione CodeDeploy con Amazon EC2 Auto Scaling Scaling, consulta.

Il tipo di implementazione

Un tipo di distribuzione è un metodo utilizzato per rendere disponibile la revisione più recente dell'applicazione sulle istanze di un gruppo di distribuzione. Esistono due tipi di distribuzione:

- In-place distribuzione: l'applicazione su ogni istanza del gruppo di distribuzione viene interrotta, viene installata la revisione più recente dell'applicazione e la nuova versione dell'applicazione viene avviata e convalidata. È possibile utilizzare un sistema di bilanciamento del carico in modo che

ogni istanza venga annullata durante la distribuzione e quindi ripristinata in servizio al termine della distribuzione. Solo le distribuzioni che utilizzano la piattaforma di EC2/On-Premises elaborazione possono utilizzare le distribuzioni sul posto. Per ulteriori informazioni sulle implementazioni sul posto, consulta. Panoramica di una distribuzione sul posto

- Blue/green implementazione: il comportamento della distribuzione dipende dalla piattaforma di elaborazione utilizzata:
  - Blue/green su una piattaforma di EC2/On-Premises elaborazione: le istanze in un gruppo di distribuzione (l'ambiente originale) vengono sostituite da un diverso set di istanze (l'ambiente sostitutivo) utilizzando questi passaggi:
    - Le istanze vengono fornite per l'ambiente sostitutivo.
    - La revisione più recente dell'applicazione viene installata sulle istanze sostitutive.
    - Si verifica un tempo di attesa opzionale per attività come il test delle applicazioni e la verifica del sistema.
    - Le istanze nell'ambiente sostitutivo vengono registrate con uno o più sistemi di bilanciamento del carico Elastic Load Balancing, causando il reindirizzamento del traffico verso di essi. Le istanze nell'ambiente originale vengono cancellate e possono essere terminate o mantenute in esecuzione per altri usi.

    > **Note**
    >
    > Se utilizzi una piattaforma di EC2/On-Premises elaborazione, tieni presente che le blue/green distribuzioni funzionano solo con istanze Amazon EC2.

  - Blue/green su una piattaforma AWS Lambda di elaborazione Amazon ECS: il traffico viene spostato in incrementi in base a una configurazione di distribuzione canaria, lineare o completa.
  - Blue/green implementazioni tramite CloudFormation: Il traffico viene spostato dalle risorse correnti alle risorse aggiornate come parte di un aggiornamento dello stack. CloudFormation Attualmente sono supportate solo le implementazioni ECS blue/green .

Per ulteriori informazioni sulle blue/green distribuzioni, consulta. Panoramica di una distribuzione blue/green

> **Note**
>
> Le blue/green implementazioni di Amazon ECS sono supportate utilizzando entrambi e. CodeDeploy CloudFormation I dettagli relativi a queste distribuzioni sono descritti nelle sezioni successive.

Profilo dell'istanza IAM

Un profilo di istanza IAM è un ruolo IAM che colleghi alle tue istanze Amazon EC2. Questo profilo include le autorizzazioni necessarie per accedere ai bucket GitHub o ai repository Amazon S3 in cui sono archiviate le applicazioni. Per ulteriori informazioni, consulta Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2.

Revision

Una revisione è una versione dell'applicazione. Una revisione della distribuzione AWS Lambda è un JSON-formatted file YAML che specifica le informazioni sulla funzione Lambda da distribuire. Una revisione della EC2/On-Premises distribuzione è un file di archivio che contiene il contenuto sorgente (codice sorgente, pagine Web, file eseguibili e script di distribuzione) e un file di specifiche dell'applicazione (file). AppSpec AWS Le revisioni Lambda possono essere archiviate in bucket Amazon S3. EC2/On-Premises le revisioni vengono archiviate in GitHub bucket o repository Amazon S3. Per Amazon S3, una revisione viene identificata in maniera univoca mediante la sua chiave oggetto Amazon S3 e il suo ETag, versione o entrambi. Infatti GitHub, una revisione è identificata in modo univoco dal relativo ID di commit.

Ruolo del servizio

Un ruolo di servizio è un ruolo IAM che concede le autorizzazioni a un AWS servizio in modo che possa accedere alle AWS risorse. Le policy associate al ruolo di servizio determinano a quali AWS risorse il servizio può accedere e le azioni che può eseguire con tali risorse. Infatti CodeDeploy, un ruolo di servizio viene utilizzato per quanto segue:

- Per leggere i tag applicati alle istanze o i nomi dei gruppi Amazon EC2 Auto Scaling associati alle istanze. Ciò consente di CodeDeploy identificare le istanze in cui può distribuire applicazioni.
- Per eseguire operazioni su istanze, gruppi Amazon EC2 Auto Scaling e sistemi di bilanciamento del carico Elastic Load Balancing.

- Pubblicare informazioni su argomenti di Amazon SNS in modo che le notifiche possano essere inviate quando si verificano eventi di distribuzione o istanza specifici.
- Per recuperare informazioni sugli CloudWatch allarmi per configurare il monitoraggio degli allarmi per le distribuzioni.

Per ulteriori informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.

Revisione dell'obiettivo

Una revisione target è la versione più recente della revisione dell'applicazione che hai caricato nel tuo repository e che desideri distribuire alle istanze di un gruppo di distribuzione. In altre parole, la revisione dell'applicazione attualmente destinata alla distribuzione. che è anche la revisione che viene utilizzata per le distribuzioni automatiche.

Altri componenti

Per informazioni sugli altri componenti del CodeDeploy flusso di lavoro, consulta i seguenti argomenti:

- Scegli un tipo di CodeDeploy repository
- Deployments
- Application Specification Files
- Instance Health
- Collaborazione con l' CodeDeploy agente
- Working with On-Premises Instances

CodeDeploy implementazioni

Questo argomento fornisce informazioni sulle componenti e sul flusso di lavoro delle distribuzioni in CodeDeploy. Il processo di distribuzione varia a seconda della piattaforma di elaborazione o del metodo di distribuzione (Lambda, Amazon ECS EC2/On-Premises o AWS CloudFormation tramite) che usi per le distribuzioni.

Argomenti

- Implementazioni su un AWS Piattaforma di elaborazione Lambda

- Implementazioni su una piattaforma di calcolo Amazon ECS
- Distribuzioni su una piattaforma di elaborazione EC2/On-Premises

Implementazioni su un AWS Piattaforma di elaborazione Lambda

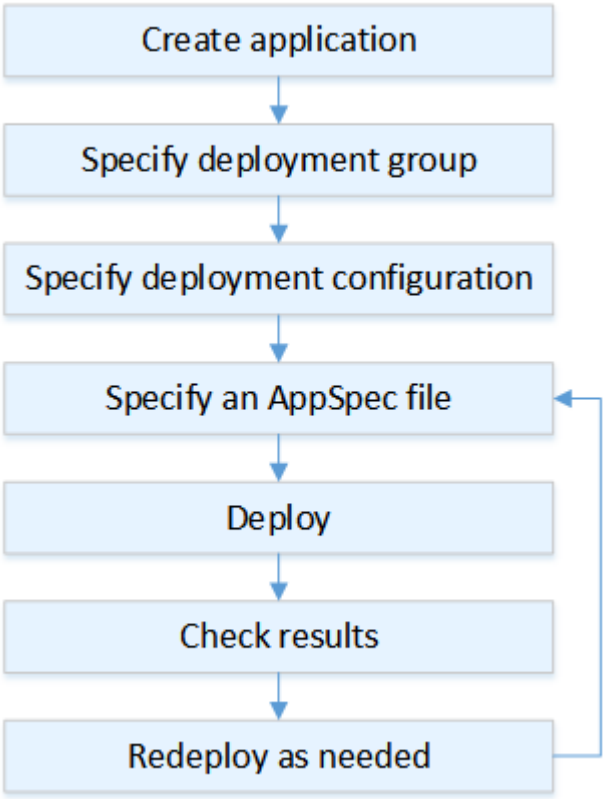
Questo argomento fornisce informazioni sui componenti e sul flusso di lavoro delle CodeDeploy distribuzioni che utilizzano la piattaforma di calcolo AWS Lambda.

Argomenti

- Flusso di lavoro di distribuzione su un AWS Lambda piattaforma di elaborazione
- Caricamento della revisione dell'applicazione
- Creazione di gruppi di applicazioni e distribuzione
- Implementazione della revisione dell'applicazione
- Aggiornamento dell'applicazione
- Distribuzioni interrotte e non riuscite
- Ridistribuzioni e rollback della distribuzione

Flusso di lavoro di distribuzione su un AWS Lambda piattaforma di elaborazione

Il seguente diagramma mostra le fasi principali nella distribuzione di funzioni AWS Lambda nuove e aggiornate.



Queste fasi includono:

1. La creazione di un'applicazione e l'assegnazione di un nome che identifichi in modo univoco le revisioni dell'applicazione da distribuire. Per distribuire le funzioni Lambda, scegli la piattaforma di calcolo AWS Lambda quando crei la tua applicazione. CodeDeploy utilizza questo nome durante una distribuzione per assicurarsi che faccia riferimento ai componenti di distribuzione corretti, come il gruppo di distribuzione, la configurazione di distribuzione e la revisione dell'applicazione. Per ulteriori informazioni, consulta Crea un'applicazione con CodeDeploy.
2. L'impostazione di un gruppo di distribuzione specificando il nome del gruppo stesso.
3. Scegli una configurazione di distribuzione per specificare in che modo il traffico viene spostato dalla versione AWS Lambda della funzione originale alla nuova versione della funzione Lambda. Per ulteriori informazioni, consulta View Deployment Configuration Details.
4. Caricamento di un file di specifiche dell'applicazione (AppSpec file) su Amazon S3. Il AppSpec file specifica una versione della funzione Lambda e le funzioni Lambda utilizzate per convalidare la distribuzione. Se non desideri creare un AppSpec file, puoi specificare una versione della funzione Lambda e le funzioni di convalida della distribuzione Lambda direttamente nella console

utilizzando YAML o JSON. Per ulteriori informazioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.

5. Implementa la revisione dell'applicazione nel gruppo di distribuzione. AWS CodeDeploy distribuisce la revisione della funzione Lambda specificata. Il traffico viene spostato sulla revisione della funzione Lambda utilizzando il file di AppSpec distribuzione scelto al momento della creazione dell'applicazione. Per ulteriori informazioni, consulta Crea una distribuzione con CodeDeploy.
6. Il controllo dei risultati della distribuzione. Per ulteriori informazioni, consulta Monitoraggio delle implementazioni in CodeDeploy.

Caricamento della revisione dell'applicazione

Inserisci un AppSpec file in Amazon S3 o inseriscilo direttamente nella console o. AWS CLI Per ulteriori informazioni, consulta Application Specification Files.

Creazione di gruppi di applicazioni e distribuzione

Un gruppo CodeDeploy di distribuzione su una piattaforma di calcolo AWS Lambda identifica una raccolta di uno o più file. AppSpec Ogni AppSpec file può distribuire una versione della funzione Lambda. Un gruppo di distribuzione definisce anche una serie di opzioni di configurazione per le distribuzioni future, ad esempio allarmi e configurazioni di rollback.

Implementazione della revisione dell'applicazione

Ora sei pronto per distribuire la revisione della funzione specificata nel AppSpec file nel gruppo di distribuzione. Puoi usare la CodeDeploy console o il comando create-deployment. Ci sono parametri che è possibile specificare per controllare la distribuzione, tra cui la revisione, il gruppo di distribuzione e la configurazione della distribuzione.

Aggiornamento dell'applicazione

È possibile aggiornare l'applicazione e quindi utilizzare la CodeDeploy console o chiamare il comando create-deployment per inviare una revisione.

Distribuzioni interrotte e non riuscite

È possibile utilizzare la CodeDeploy console o il comando stop-deployment per interrompere una distribuzione. Quando si tenta di arrestare la distribuzione, accade una di queste tre cose:

- La distribuzione si arresta e l'operazione restituisce uno stato di successo. In questo caso, non vengono più eseguiti eventi del ciclo di vita della distribuzione sul gruppo di distribuzione per la distribuzione interrotta.
- La distribuzione non si arresta immediatamente e l'operazione restituisce uno stato in attesa. In questo caso, alcuni eventi del ciclo di vita di distribuzione potrebbe essere ancora in esecuzione sul gruppo di distribuzione. Una volta completata l'operazione in attesa, le chiamate successive per fermare la distribuzione restituiscono uno stato di successo.
- La distribuzione non è in grado di arrestarsi e l'operazione restituisce un errore. Per ulteriori informazioni, consulta ErrorInformationErrori comuni nell' AWS CodeDeploy API Reference.

Come per le implementazioni arrestate, la distribuzione non riuscita può causare l'esecuzione di alcuni eventi del ciclo di vita di distribuzione. Per scoprire perché una distribuzione non è riuscita, è possibile utilizzare la console CodeDeploy o analizzare i dati del file di log dalla distribuzione non riuscita. Per ulteriori informazioni, consultare Revisione dell'applicazione e pulizia dei file di registro e Visualizza i dati di registro per le CodeDeploy EC2/On-Premises distribuzioni.

Ridistribuzioni e rollback della distribuzione

CodeDeploy implementa i rollback ridistribuendo, come nuova distribuzione, una revisione precedentemente distribuita.

Puoi configurare un gruppo di distribuzione in modo che esegua automaticamente il rollback della distribuzione quando si soddisfano determinate condizioni, incluso quando una distribuzione non ha successo o viene raggiunta una soglia di monitoraggio degli allarmi. È inoltre possibile sostituire le impostazioni di rollback specificati per un gruppo di distribuzione in una singola distribuzione.

È inoltre possibile scegliere di eseguire il rollback di una distribuzione non riuscita ridistribuendo manualmente una revisione distribuita in precedenza.

In tutti i casi, la distribuzione nuova o su cui è stato eseguito un rollback viene assegnato il relativo ID della distribuzione. L'elenco delle distribuzioni che è possibile visualizzare nella CodeDeploy console mostra quali sono il risultato di una distribuzione automatica.

Per ulteriori informazioni, consulta Ridistribuisci e ripristina una distribuzione con CodeDeploy.

Implementazioni su una piattaforma di calcolo Amazon ECS

Questo argomento fornisce informazioni sui componenti e sul flusso di lavoro delle CodeDeploy distribuzioni che utilizzano la piattaforma di calcolo Amazon ECS.

Argomenti

- Prima di iniziare una distribuzione Amazon ECS
- Flusso di lavoro di implementazione (alto livello) su una piattaforma di calcolo Amazon ECS
- Cosa succede durante una distribuzione di Amazon ECS
- Caricamento della revisione dell'applicazione
- Creazione di gruppi di applicazioni e distribuzione
- Implementazione della revisione dell'applicazione
- Aggiornamento dell'applicazione
- Distribuzioni interrotte e non riuscite
- Ridistribuzioni e rollback della distribuzione
- blue/green Implementazioni Amazon ECS tramite AWS CloudFormation

Prima di iniziare una distribuzione Amazon ECS

Prima di iniziare la distribuzione di un'applicazione Amazon ECS, è necessario disporre di quanto segue. Alcuni requisiti vengono specificati al momento della creazione del gruppo di distribuzione, mentre altri vengono specificati nel AppSpec file.

Requisito	Specificato nel
Cluster Amazon ECS	Gruppo di distribuzione
Servizio Amazon ECS	Gruppo di distribuzione
Application Load Balancer o Network Load Balancer	Gruppo di distribuzione
Listener di produzione	Gruppo di distribuzione
Listener di test (opzionale)	Gruppo di distribuzione
Due gruppi target	Gruppo di distribuzione
Definizione dell'attività di Amazon ECS	AppSpec file
Nome del container	AppSpec fascicolo

Requisito	Specificato nel
Porta del container	AppSpec fascicolo

Cluster Amazon ECS

Un cluster Amazon ECS è un raggruppamento logico di processi o servizi. Specifichi il cluster Amazon ECS che contiene il servizio Amazon ECS quando crei il gruppo di distribuzione dell' CodeDeploy applicazione. Per ulteriori informazioni, consulta i cluster Amazon ECS nella Guida per l'utente di Amazon Elastic Container Service.

Servizio Amazon ECS

Un servizio Amazon ECS mantiene ed esegue istanze specifiche di una definizione di attività in un cluster Amazon ECS. Il tuo servizio Amazon ECS deve essere abilitato per CodeDeploy. Per impostazione predefinita, un servizio Amazon ECS è abilitato per le distribuzioni Amazon ECS. Quando crei il tuo gruppo di distribuzione, scegli di distribuire un servizio Amazon ECS che si trova nel tuo cluster Amazon ECS. Per ulteriori informazioni, consulta i servizi Amazon ECS nella Guida per l'utente di Amazon Elastic Container Service.

Application Load Balancer o Network Load Balancer

È necessario utilizzare Elastic Load Balancing con il servizio Amazon ECS che si desidera aggiornare con una distribuzione Amazon ECS. È possibile utilizzare un Application Load Balancer o un Network Load Balancer. Ti consigliamo un Application Load Balancer per sfruttare funzionalità come la mappatura dinamica delle porte e le regole di routing e priorità basate sui percorsi. Il load balancer viene specificato quando si crea il gruppo di distribuzione dell'applicazione CodeDeploy . Per ulteriori informazioni, consulta Configura un sistema di bilanciamento del carico, gruppi target e listener per le distribuzioni di CodeDeploy Amazon ECS la sezione Creazione di un bilanciamento del carico nella Guida per l'utente di Amazon Elastic Container Service.

Uno o due ascoltatori

Un listener viene utilizzato dal sistema di bilanciamento del carico per indirizzare il traffico verso i tuoi gruppi target. Un listener di produzione è obbligatorio. Puoi specificare un secondo listener di test opzionale che indirizza il traffico verso il set di attività di sostituzione mentre esegui test di convalida. Puoi specificare uno o entrambi i listener al momento della creazione di un gruppo di distribuzione. Se usi la console Amazon ECS per creare il tuo servizio Amazon ECS, i tuoi listener vengono creati per te. Per ulteriori informazioni, consulta Listeners for your application

load balancer nella Elastic Load Balancing User Guide e Creazione di un servizio nella Amazon Elastic Container Service User Guide.

Due gruppi target di Amazon ECS

Un gruppo target viene utilizzato per instradare il traffico verso un target registrato. Una distribuzione di Amazon ECS richiede due gruppi target: uno per il set di attività originale dell'applicazione Amazon ECS e uno per il set di attività sostitutivo. Durante la distribuzione, CodeDeploy crea un set di attività sostitutivo e reindirizza il traffico dal set di attività originale a quello nuovo. Devi specificare i gruppi target al momento di creare un gruppo di distribuzione dell'applicazione CodeDeploy .

Durante una distribuzione, CodeDeploy determina quale gruppo target è associato all'attività impostata nel servizio Amazon ECS con lo stato `PRIMARY` (questo è il set di attività originale), associa un gruppo target ad esso, quindi associa l'altro gruppo target al set di attività sostitutivo. Se effettui un'altra distribuzione, il gruppo target associato al set di attività originale della distribuzione attuale viene associato al set di attività di sostituzione della distribuzione successiva. Per ulteriori informazioni, consulta Target groups for your application load balancer nella Elastic Load Balancing User Guide.

Una definizione di attività Amazon ECS

È necessaria una definizione di attività per eseguire il contenitore Docker che contiene l'applicazione Amazon ECS. L'ARN della definizione dell'attività viene specificato nel file dell' CodeDeploy AppSpec applicazione. Per ulteriori informazioni, consulta le definizioni delle attività di Amazon ECS nella Guida per l'utente di Amazon Elastic Container Service eAppSpec sezione «risorse» per le distribuzioni di Amazon ECS.

Un contenitore per la tua applicazione Amazon ECS

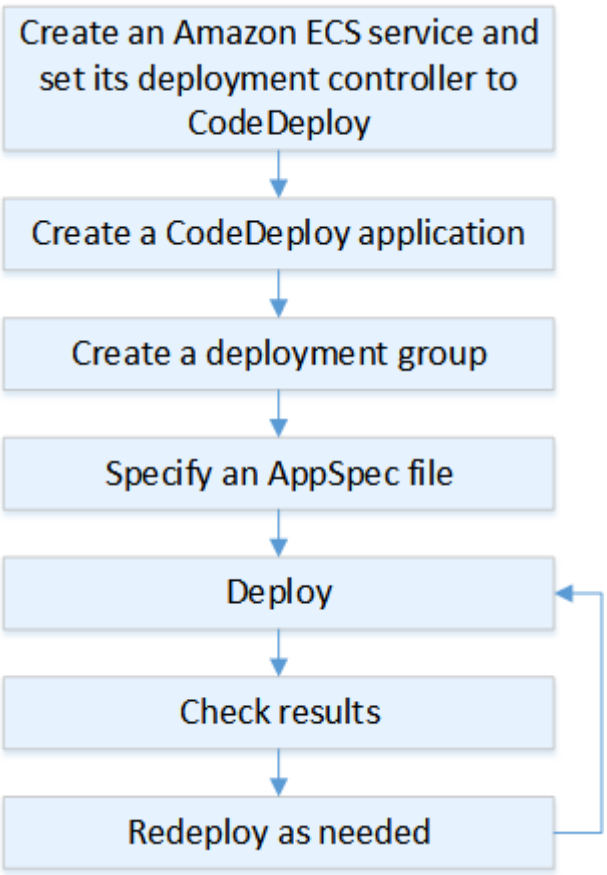
Un container Docker è un'unità di software che include il codice e le sue dipendenze in modo che l'applicazione possa essere eseguita. Un container isola l'applicazione in modo che funzioni in diversi ambienti di elaborazione. Il sistema di bilanciamento del carico indirizza il traffico verso un contenitore nel set di attività dell'applicazione Amazon ECS. Specificate il nome del contenitore nel file CodeDeploy dell'applicazione. AppSpec Il contenitore specificato nel AppSpec file deve essere uno dei contenitori specificati nella definizione dell'attività Amazon ECS. Per ulteriori informazioni, consulta Cos'è Amazon Elastic Container Service? nella Guida per l'utente di Amazon Elastic Container Service eAppSpec sezione «risorse» per le distribuzioni di Amazon ECS.

Una porta per il set di attività sostitutivo

Durante la distribuzione di Amazon ECS, il sistema di bilanciamento del carico indirizza il traffico verso questa porta sul contenitore specificato nel file CodeDeploy dell'applicazione. AppSpec Devi specificare la porta nel file CodeDeploy dell'applicazione. AppSpec Per ulteriori informazioni, consulta AppSpec sezione «risorse» per le distribuzioni di Amazon ECS.

Flusso di lavoro di implementazione (alto livello) su una piattaforma di calcolo Amazon ECS

Il diagramma seguente mostra le fasi principali della distribuzione di servizi Amazon ECS aggiornati.



Queste fasi includono:

1. Crea un' AWS CodeDeploy applicazione specificando un nome che rappresenti in modo univoco ciò che desideri distribuire. Per distribuire un'applicazione Amazon ECS, nella tua AWS CodeDeploy applicazione, scegli la piattaforma di calcolo Amazon ECS. CodeDeploy utilizza un'applicazione durante una distribuzione per fare riferimento ai componenti di distribuzione corretti, come il gruppo di distribuzione, i gruppi target, i listener, il comportamento

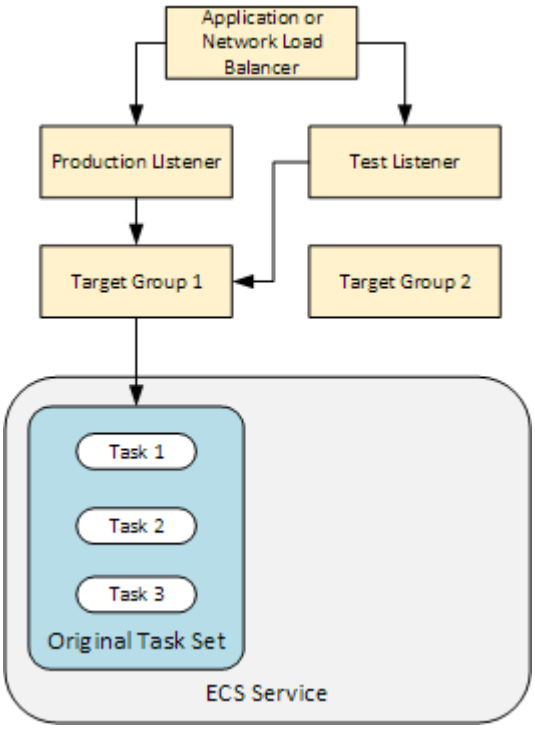
di reindirizzamento del traffico e la revisione dell'applicazione. Per ulteriori informazioni, consulta Crea un'applicazione con CodeDeploy.

2. La creazione di un gruppo di distribuzione specificando:
   - Il nome del gruppo di distribuzione.
   - Il nome del cluster e del servizio Amazon ECS. Il controller di distribuzione del servizio Amazon ECS deve essere impostato CodeDeploy su.
   - Il listener di produzione, un listener di test opzionale e i gruppi target utilizzati durante una distribuzione.
   - Impostazioni di distribuzione, ad esempio quando reindirizzare il traffico di produzione verso l'attività Amazon ECS sostitutiva impostata nel servizio Amazon ECS e quando terminare l'attività Amazon ECS originale impostata nel servizio Amazon ECS.
   - Impostazioni opzionali, ad esempio trigger, allarmi e comportamento di rollback.
3. Specificare un file (file) con le specifiche dell'applicazione. AppSpec Puoi caricarlo su Amazon S3, inserirlo nella console in formato YAML o JSON o specificarlo con o SDK. AWS CLI Il AppSpec file specifica una definizione di attività Amazon ECS per la distribuzione, una mappatura del nome del contenitore e delle porte utilizzata per instradare il traffico e le funzioni Lambda eseguite dopo gli hook del ciclo di vita della distribuzione. Il nome del contenitore deve essere un contenitore nella definizione dell'attività Amazon ECS. Per ulteriori informazioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.
4. Implementa la revisione dell'applicazione. AWS CodeDeploy reindirizza il traffico dalla versione originale di un set di attività nel tuo servizio Amazon ECS verso un nuovo set di attività sostitutivo. I gruppi target specificati nel gruppo di distribuzione vengono utilizzati per instradare il traffico verso il set di attività originale e di sostituzione. Una volta completata la distribuzione, il set di attività originale viene terminato. Puoi specificare un listener di test opzionale per inoltrare il traffico di test verso la versione di sostituzione prima che il traffico venga reinstradato. Per ulteriori informazioni, consulta Crea una distribuzione con CodeDeploy.
5. Il controllo dei risultati della distribuzione. Per ulteriori informazioni, consulta Monitoraggio delle implementazioni in CodeDeploy.

Cosa succede durante una distribuzione di Amazon ECS

Prima di iniziare una distribuzione Amazon ECS con un listener di test, devi configurarne i componenti. Per ulteriori informazioni, consulta Prima di iniziare una distribuzione Amazon ECS.

Il diagramma seguente mostra la relazione tra questi componenti quando una distribuzione Amazon ECS è pronta per l'avvio.



All'avvio della distribuzione, gli eventi del ciclo di vita della distribuzione iniziano a essere eseguiti uno alla volta. Alcuni eventi del ciclo di vita sono hook che eseguono solo le funzioni Lambda specificate nel file. AppSpec Gli eventi del ciclo di vita della distribuzione nella seguente tabella sono elencati nell'ordine in cui vengono eseguiti. Per ulteriori informazioni, consulta AppSpec sezione 'hook' per una distribuzione Amazon ECS.

Evento del ciclo di vita	Azione relativa all'evento del ciclo di vita
`BeforeInstall` (un gancio per le funzioni Lambda)	Esegui le funzioni Lambda.
Installa	Configurare il set di attività di sostituzione.
`AfterInstall` (un gancio per le funzioni Lambda)	Esegui le funzioni Lambda.
AllowTestTraffic	Instradare il traffico dal listener di test al gruppo target 2.

Evento del ciclo di vita	Azione relativa all'evento del ciclo di vita
`AfterAllowTestTraffic` (un gancio per le funzioni Lambda)	Esegui le funzioni Lambda.
`BeforeAllowTraffic` (un gancio per le funzioni Lambda)	Esegui le funzioni Lambda.
AllowTraffic	Instradare il traffico listener di produzione al gruppo target 2.
`AfterAllowTraffic`	Esegui le funzioni Lambda.

> **Note**
>
> Le funzioni Lambda in un hook sono opzionali.

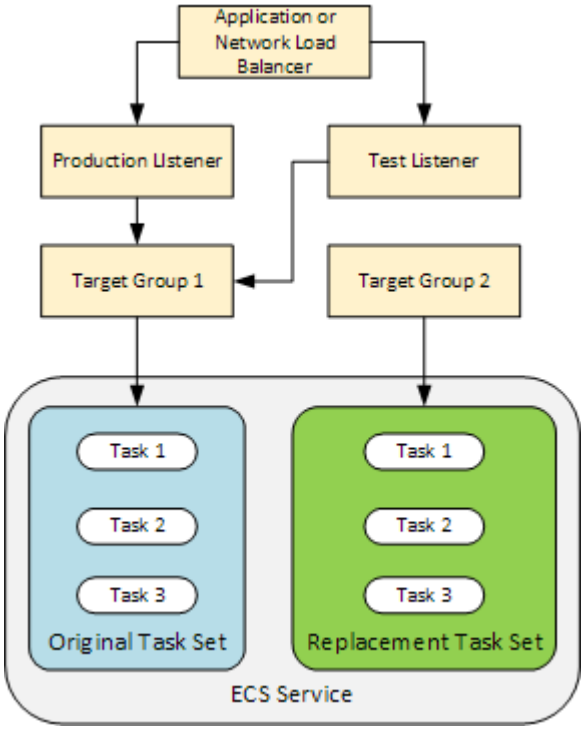
1. Esegui tutte le funzioni Lambda specificate nell'`BeforeInstall`hook del file. AppSpec
2. Durante l'evento del ciclo di vita `Install`:

   a. Nel tuo servizio Amazon ECS viene creato un set di attività sostitutivo.

   b. Viene installata l'applicazione containerizzata aggiornata nel set di attività di sostituzione.

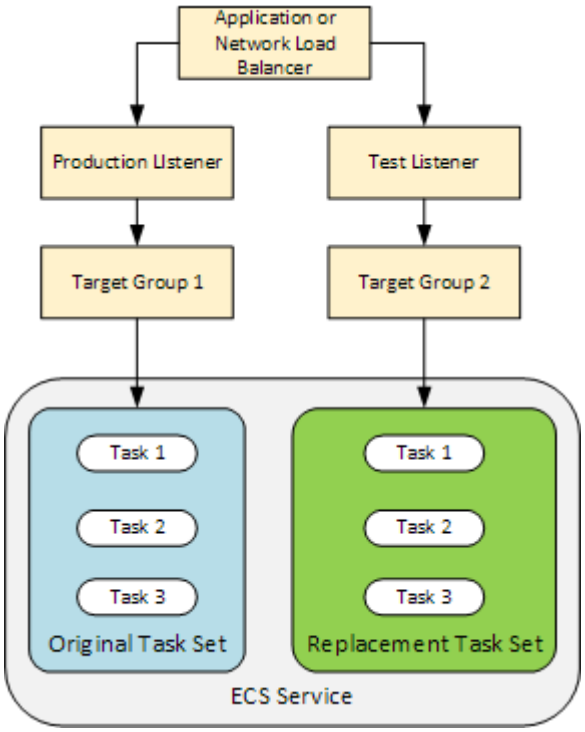
   c. Viene associato il secondo gruppo target al set di attività di sostituzione.

   Questo diagramma mostra i componenti di distribuzione con il nuovo set di attività di sostituzione. L'applicazione containerizzata è all'interno del set di attività. Il set di attività è composto da tre attività. Un'applicazione può avere un numero illimitato di attività. Il secondo gruppo target è ora associato al set di attività di sostituzione.



3. Esegui tutte le funzioni Lambda specificate nell'`AfterInstall`hook del file. AppSpec

4. Viene richiamato l'evento `AllowTestTraffic`. Durante questo evento del ciclo di vita, il listener di test instrada il traffico all'applicazione containerizzata aggiornata.



5. Esegui tutte le funzioni Lambda specificate nell'`AfterAllowTestTraffic`hook del file. AppSpec Le funzioni Lambda possono convalidare l'implementazione utilizzando il traffico di test. Ad esempio, una funzione Lambda può fornire traffico al listener di test e tracciare i parametri dal set di attività di sostituzione. Se i rollback sono configurati, puoi configurare un CloudWatch allarme che attiva un rollback quando il test di convalida nella funzione Lambda fallisce.

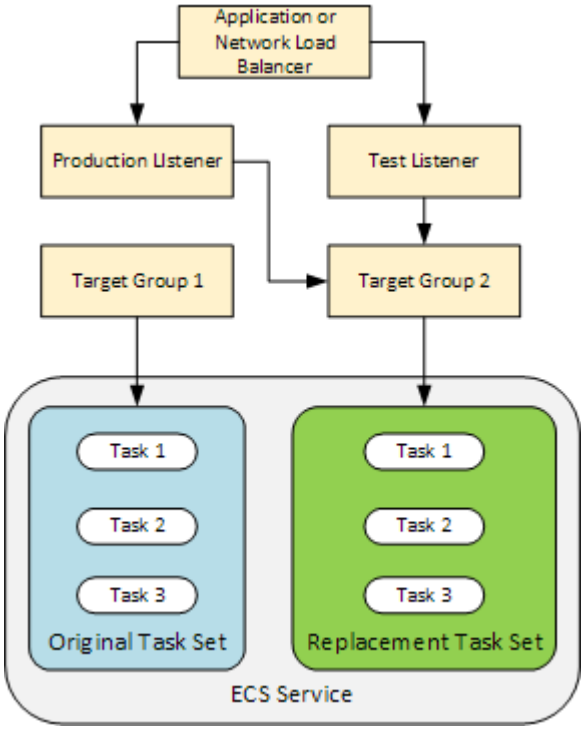
   Una volta completati i test di convalida, si verifica una delle seguenti condizioni:

   - Se la convalida non riesce e sono configurati i rollback, lo stato della distribuzione viene contrassegnato `Failed` e i componenti vengono ripristinati allo stato di avvio della distribuzione.
   - Se la convalida non riesce e non sono configurati i rollback, lo stato della distribuzione viene contrassegnato `Failed` e i componenti rimangono nello stato corrente.
   - Se la convalida riesce, la distribuzione continua all'hook `BeforeAllowTraffic`.

   Per ulteriori informazioni, consultare Monitoraggio delle distribuzioni con allarmi in CloudWatch CodeDeploy, Rollback automatici e Configurazione delle opzioni avanzate per un gruppo di distribuzione.

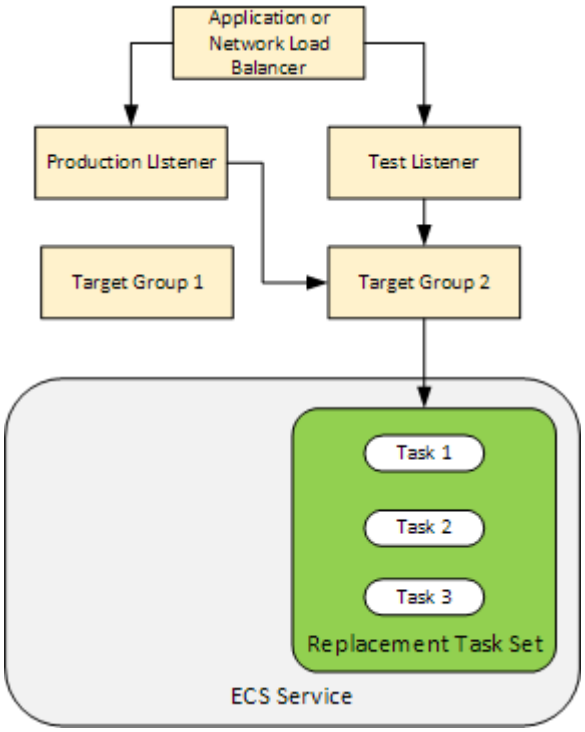
6. Esegui tutte le funzioni Lambda specificate nell'`BeforeAllowTraffic`hook del file. AppSpec

7. Viene richiamato l'evento `AllowTraffic`. Il traffico di produzione viene reinstradato dal set di attività originale al set di attività di sostituzione. Il seguente diagramma mostra il set di attività di sostituzione che riceve il traffico di produzione.



8. Esegui tutte le funzioni Lambda specificate nell'`AfterAllowTraffic`hook del file. AppSpec

9. All'esito positivo di tutti gli eventi, lo stato della distribuzione viene impostato su `Succeeded` e il set di attività originale viene rimosso.



Caricamento della revisione dell'applicazione

Inserisci un AppSpec file in Amazon S3 o inseriscilo direttamente nella console o. AWS CLI Per ulteriori informazioni, consulta Application Specification Files.

Creazione di gruppi di applicazioni e distribuzione

Un gruppo di CodeDeploy distribuzione su una piattaforma di calcolo Amazon ECS identifica i listener per servire il traffico verso l'applicazione Amazon ECS aggiornata e due gruppi target utilizzati durante la distribuzione. Un gruppo di distribuzione definisce anche un set di opzioni di configurazione, ad esempio allarmi e configurazioni di rollback.

Implementazione della revisione dell'applicazione

Ora sei pronto per distribuire il servizio Amazon ECS aggiornato specificato nel tuo gruppo di distribuzione. Puoi usare la CodeDeploy console o il comando create-deployment. Per controllare la distribuzione puoi specificare parametri quali la revisione e il gruppo di distribuzione.

Aggiornamento dell'applicazione

È possibile aggiornare l'applicazione e quindi utilizzare la CodeDeploy console o chiamare il comando create-deployment per inviare una revisione.

Distribuzioni interrotte e non riuscite

È possibile utilizzare la CodeDeploy console o il comando stop-deployment per interrompere una distribuzione. Quando si tenta di arrestare la distribuzione, accade una di queste tre cose:

- La distribuzione si arresta e l'operazione restituisce uno stato di successo. In questo caso, non vengono più eseguiti eventi del ciclo di vita della distribuzione sul gruppo di distribuzione per la distribuzione interrotta.
- La distribuzione non si arresta immediatamente e l'operazione restituisce uno stato in attesa. In questo caso, alcuni eventi del ciclo di vita di distribuzione potrebbe essere ancora in esecuzione sul gruppo di distribuzione. Una volta completata l'operazione in attesa, le chiamate successive per fermare la distribuzione restituiscono uno stato di successo.
- La distribuzione non è in grado di arrestarsi e l'operazione restituisce un errore. Per ulteriori informazioni, consulta Informazioni sugli errori ed Errori comuni nell' AWS CodeDeploy API Reference.

Ridistribuzioni e rollback della distribuzione

CodeDeploy implementa i rollback reindirizzando il traffico dal set di attività sostitutivo al set di attività originale.

Puoi configurare un gruppo di distribuzione in modo che esegua automaticamente il rollback della distribuzione quando si soddisfano determinate condizioni, incluso quando una distribuzione non ha successo o viene raggiunta una soglia di monitoraggio degli allarmi. È inoltre possibile sostituire le impostazioni di rollback specificati per un gruppo di distribuzione in una singola distribuzione.

È inoltre possibile scegliere di eseguire il rollback di una distribuzione non riuscita ridistribuendo manualmente una revisione distribuita in precedenza.

In tutti i casi, la distribuzione nuova o su cui è stato eseguito un rollback viene assegnato il relativo ID della distribuzione. La console CodeDeploy mostra un elenco di distribuzioni frutto di una distribuzione automatica.

Se effettui una nuova distribuzione, il gruppo target associato al set di attività originale della distribuzione attuale viene associato al set di attività di sostituzione della nuova distribuzione.

Per ulteriori informazioni, consulta Ridistribuisci e ripristina una distribuzione con CodeDeploy.

blue/green Implementazioni Amazon ECS tramite AWS CloudFormation

Puoi utilizzarlo AWS CloudFormation per gestire le blue/green distribuzioni di Amazon ECS tramite. CodeDeploy Per ulteriori informazioni, consulta Crea una blue/green distribuzione Amazon ECS tramite CloudFormation.

> **Note**
>
> La gestione delle blue/green distribuzioni Amazon ECS con non CloudFormation è disponibile nella regione Asia Pacifico (Osaka).

Distribuzioni su una piattaforma di elaborazione EC2/On-Premises

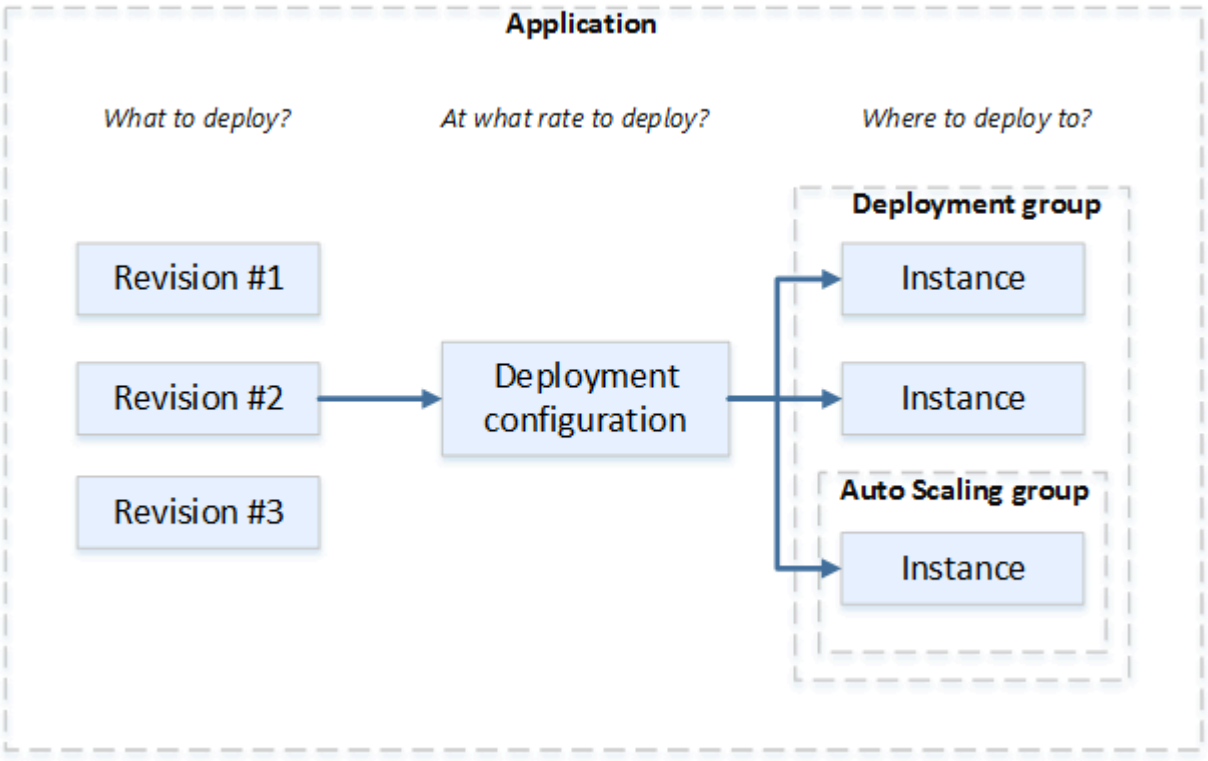
Questo argomento fornisce informazioni sui componenti e sul flusso di lavoro delle CodeDeploy distribuzioni che utilizzano la piattaforma di elaborazione. EC2/On-Premises Per informazioni sulle blue/green distribuzioni, consulta. Panoramica di una distribuzione blue/green

Argomenti

- Componenti di distribuzione su una piattaforma di elaborazione EC2/on-premises
- Flusso di lavoro di implementazione su una piattaforma di EC2/on-premises elaborazione
- Configurazione delle istanze
- Caricamento della revisione dell'applicazione
- Creazione dei gruppi di applicazioni e di distribuzione
- Implementazione della revisione dell'applicazione
- Aggiornamento dell'applicazione
- Distribuzioni interrotte e non riuscite
- Ridistribuzioni e rollback di distribuzione

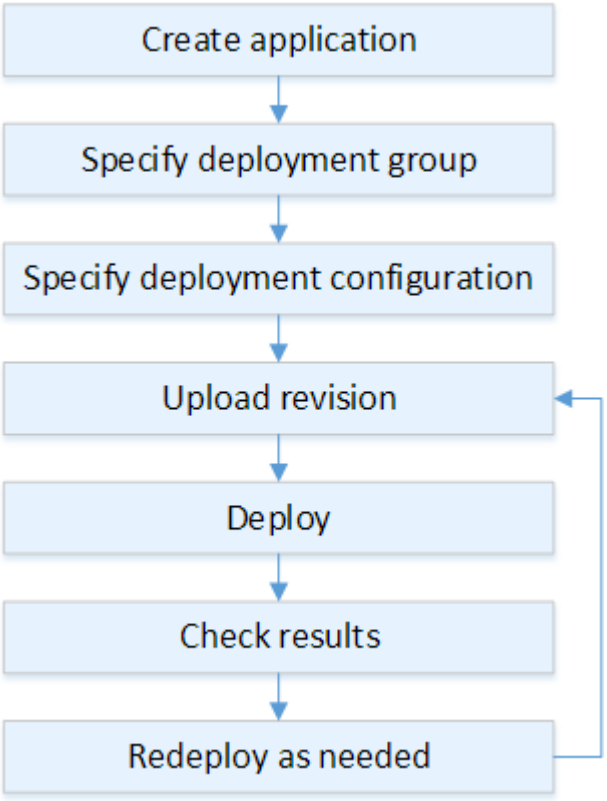
Componenti di distribuzione su una piattaforma di elaborazione EC2/on-premises

Il diagramma seguente mostra i componenti di una CodeDeploy distribuzione su una piattaforma di EC2/On-Premises elaborazione.



Flusso di lavoro di implementazione su una piattaforma di EC2/on-premises elaborazione

Il seguente diagramma mostra le fasi principali nella distribuzione di revisioni dell'applicazione:



Queste fasi includono:

1. Crea un'applicazione e assegnale un nome che identifichi in modo univoco le revisioni dell'applicazione che desideri distribuire e la piattaforma di elaborazione per la tua applicazione. CodeDeploy utilizza questo nome durante una distribuzione per assicurarsi che faccia riferimento ai componenti di distribuzione corretti, come il gruppo di distribuzione, la configurazione di distribuzione e la revisione dell'applicazione. Per ulteriori informazioni, consulta Crea un'applicazione con CodeDeploy.
2. L'impostazione di un gruppo di distribuzione specificando un tipo di distribuzione e le istanze in cui si desidera distribuire le revisioni dell'applicazione. Un'implementazione in loco aggiorna le istanze alla revisione dell'applicazione più recente. Una blue/green distribuzione registra un set sostitutivo di istanze per il gruppo di distribuzione con un sistema di bilanciamento del carico e annulla la registrazione delle istanze originali.

   Puoi specificare i tag applicati alle istanze, i nomi dei gruppi Amazon EC2 Auto Scaling o entrambi.

   Se specifichi un gruppo di tag in un gruppo di distribuzione, CodeDeploy esegui la distribuzione su istanze a cui è applicato almeno uno dei tag specificati. Se specifichi due o più gruppi di tag,

esegue la CodeDeploy distribuzione solo nelle istanze che soddisfano i criteri per ciascuno dei gruppi di tag. Per ulteriori informazioni, consulta Tagging Instances for Deployments.

In tutti i casi, le istanze devono essere configurate per essere utilizzate in una distribuzione (ovvero devono essere etichettate o appartenere a un gruppo Amazon EC2 Auto Scaling) e avere CodeDeploy l'agente installato e funzionante.

Puoi utilizzare un CloudFormation modello per configurare rapidamente un'istanza Amazon EC2 basata su Amazon Linux o Windows Server. Ti forniamo anche l' CodeDeploy agente autonomo in modo che tu possa installarlo su istanze di Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) o Windows Server. Per ulteriori informazioni, consulta Crea un gruppo di distribuzione con CodeDeploy.

È anche possibile specificare le seguenti opzioni:

- Notifiche Amazon SNS. Crea trigger che inviano notifiche agli abbonati di un argomento di Amazon SNS quando si verificano eventi specifici, come eventi di successo o di fallimento, nelle distribuzioni e nelle istanze. Per ulteriori informazioni, consulta Monitoring Deployments with Amazon SNS Event Notifications.
- Alarm-based gestione della distribuzione. Implementa il monitoraggio degli CloudWatch allarmi di Amazon per interrompere le implementazioni quando i parametri superano o scendono al di sotto delle soglie impostate. CloudWatch
- Rollback automatici di distribuzione Configura una distribuzione per eseguire automaticamente il rollback alla revisione precedentemente nota come correttamente funzionante quando una distribuzione non riesce o viene raggiunta una soglia di allarme.

3. La specifica di una configurazione di distribuzione per indicare a quante istanze le revisioni delle applicazioni devono essere distribuite simultaneamente e descrivere le condizioni di successo e di errore per la distribuzione. Per ulteriori informazioni, consulta View Deployment Configuration Details.
4. Carica una revisione dell'applicazione su Amazon GitHub S3 o. Oltre ai file che desideri distribuire e agli script che desideri eseguire durante la distribuzione, devi includere un file (file) con le specifiche dell'applicazione. AppSpec Questo file contiene le istruzioni di distribuzione, ad esempio il percorso in cui copiare i file su ogni istanza e quando eseguire lo script di distribuzione. Per ulteriori informazioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.
5. La distribuzione della revisione dell'applicazione al gruppo di distribuzione. L' CodeDeploy agente su ogni istanza del gruppo di distribuzione copia la revisione dell'applicazione da Amazon S3 GitHub o sull'istanza. L' CodeDeploy agente quindi separa la revisione e, utilizzando il AppSpec

file, copia i file nelle posizioni specificate ed esegue tutti gli script di distribuzione. Per ulteriori informazioni, consulta Crea una distribuzione con CodeDeploy.

6. Il controllo dei risultati della distribuzione. Per ulteriori informazioni, consulta Monitoraggio delle implementazioni in CodeDeploy.
7. La ridistribuzione di una revisione. È possibile eseguire questa operazione se hai bisogno di correggere un bug nel contenuto sorgente o eseguire gli script di distribuzione in un ordine diverso o inviare una distribuzione non riuscita. A tale scopo, raggruppa il contenuto sorgente modificato, gli eventuali script di distribuzione e il AppSpec file in una nuova revisione, quindi carica la revisione nel bucket o nell'archivio Amazon S3. GitHub Esegui quindi una nuova distribuzione per lo stesso gruppo di distribuzione con la nuova revisione. Per ulteriori informazioni, consulta Crea una distribuzione con CodeDeploy.

Configurazione delle istanze

È necessario configurare le istanze prima di poter distribuire le revisioni dell'applicazione per la prima volta. Se una revisione di un'applicazione richiede tre server di produzione e due server di backup, puoi lanciare o utilizzare cinque istanze.

Per effettuare manualmente il provisioning delle istanze:

1. Installa l' CodeDeploy agente sulle istanze. L' CodeDeploy agente può essere installato su istanze di Amazon Linux, Ubuntu Server, RHEL e Windows Server.
2. Abilita i tag, se utilizzi i tag per identificare le istanze in un gruppo di distribuzione. CodeDeploy si basa sui tag per identificare e raggruppare le istanze in gruppi di distribuzione. CodeDeploy Sebbene nel tutorial Nozioni di base siano utilizzati entrambi, è sufficiente utilizzare una chiave o un valore per definire un tag per un gruppo di distribuzione.
3. Avvia istanze Amazon EC2 con un profilo di istanza IAM collegato. Il profilo dell'istanza IAM deve essere collegato a un'istanza Amazon EC2 non appena viene avviata per consentire all' CodeDeploy agente di verificare l'identità dell'istanza.
4. Creazione di un ruolo del servizio Fornisci l'accesso al servizio in modo da CodeDeploy poter espandere i tag nel tuo AWS account.

Per una distribuzione iniziale, il CloudFormation modello fa tutto questo per te. Crea e configura nuove istanze singole di Amazon EC2 basate su Amazon Linux o Windows Server con CodeDeploy l'agente già installato. Per ulteriori informazioni, consulta Utilizzo delle istanze per CodeDeploy.

> **Note**
>
> Per una blue/green distribuzione, puoi scegliere se utilizzare le istanze che già possiedi per l'ambiente sostitutivo o consentire il CodeDeploy provisioning di nuove istanze come parte del processo di distribuzione.

Caricamento della revisione dell'applicazione

Inserite un AppSpec file nella cartella principale nella struttura delle cartelle del contenuto di origine dell'applicazione. Per ulteriori informazioni, consulta Application Specification Files.

Aggrega la struttura della cartella del contenuto sorgente dell'applicazione in un formato di file di archivio come zip, tar o tar compresso. Carica il file di archivio (la revisione) in un GitHub bucket o repository Amazon S3.

> **Note**
>
> I formati di file di archivio tar e tar compressi (.tar e .tar.gz) non sono supportati per le istanze di Windows Server.

Creazione dei gruppi di applicazioni e di distribuzione

Un gruppo CodeDeploy di distribuzione identifica una raccolta di istanze in base ai relativi tag, ai nomi dei gruppi Amazon EC2 Auto Scaling o a entrambi. Nella stessa istanza possono essere implementate più revisioni dell'applicazione. Una revisione di un'applicazione può essere implementata in più istanze.

Ad esempio, è possibile aggiungere un tag di "Prod" per i tre server di produzione e "Backup" per i due server di backup. Questi due tag possono essere utilizzati per creare due diversi gruppi di distribuzione nell' CodeDeploy applicazione, consentendoti di scegliere quale set di server (o entrambi) deve partecipare a una distribuzione.

È possibile utilizzare diversi gruppi di tag in un gruppo di distribuzione per limitare le distribuzioni a un set di istanze più piccolo. Per informazioni, consulta Tagging Instances for Deployments.

Implementazione della revisione dell'applicazione

Ora sei pronto per distribuire la revisione dell'applicazione da Amazon S3 GitHub o al gruppo di distribuzione. Puoi usare la CodeDeploy console o il comando create-deployment. Ci sono parametri che è possibile specificare per controllare la distribuzione, tra cui la revisione, il gruppo di distribuzione e la configurazione della distribuzione.

Aggiornamento dell'applicazione

È possibile aggiornare l'applicazione e quindi utilizzare la CodeDeploy console o chiamare il comando create-deployment per inviare una revisione.

Distribuzioni interrotte e non riuscite

È possibile utilizzare la CodeDeploy console o il comando stop-deployment per interrompere una distribuzione. Quando si tenta di arrestare la distribuzione, accade una di queste tre cose:

- La distribuzione si arresta e l'operazione restituisce uno stato di successo. In questo caso, non vengono più eseguiti eventi del ciclo di vita della distribuzione sul gruppo di distribuzione per la distribuzione interrotta. Alcuni file potrebbero essere già stati copiati e alcuni script potrebbero essere già stati eseguiti su una o più istanze nel gruppo di distribuzione.
- La distribuzione non si arresta immediatamente e l'operazione restituisce uno stato in attesa. In questo caso, alcuni eventi del ciclo di vita di distribuzione potrebbe essere ancora in esecuzione sul gruppo di distribuzione. Alcuni file potrebbero essere già stati copiati e alcuni script potrebbero essere già stati eseguiti su una o più istanze nel gruppo di distribuzione. Una volta completata l'operazione in attesa, le chiamate successive per fermare la distribuzione restituiscono uno stato di successo.
- La distribuzione non è in grado di arrestarsi e l'operazione restituisce un errore. Per ulteriori informazioni, consulta ErrorInformationErrori comuni nell' AWS CodeDeploy API Reference.

Come per le distribuzioni arrestate, le distribuzioni non riuscite possono determinare l'esecuzione di alcuni eventi del ciclo di vita della distribuzione su una o più istanze nel gruppo di distribuzione. Per scoprire perché una distribuzione non è riuscita, puoi utilizzare la CodeDeploy console, chiamare il comando get-deployment-instance o analizzare i dati del file di registro relativi alla distribuzione non riuscita. Per ulteriori informazioni, consultare Revisione dell'applicazione e pulizia dei file di registro e Visualizza i dati di registro per le CodeDeploy EC2/On-Premises distribuzioni.

Ridistribuzioni e rollback di distribuzione

CodeDeploy implementa i rollback ridistribuendo, come nuova distribuzione, una revisione precedentemente distribuita.

Puoi configurare un gruppo di distribuzione in modo che esegua automaticamente il rollback della distribuzione quando si soddisfano determinate condizioni, incluso quando una distribuzione non ha successo o viene raggiunta una soglia di monitoraggio degli allarmi. È inoltre possibile sostituire le impostazioni di rollback specificati per un gruppo di distribuzione in una singola distribuzione.

È inoltre possibile scegliere di eseguire il rollback di una distribuzione non riuscita ridistribuendo manualmente una revisione distribuita in precedenza.

In tutti i casi, la distribuzione nuova o su cui è stato eseguito un rollback viene assegnato il relativo ID della distribuzione. L'elenco delle distribuzioni che è possibile visualizzare nella CodeDeploy console mostra quali sono il risultato di una distribuzione automatica.

Per ulteriori informazioni, consulta Ridistribuisci e ripristina una distribuzione con CodeDeploy.

CodeDeploy file delle specifiche dell'applicazione (AppSpec)

Un file (AppSpec file) delle specifiche dell'applicazione, che è unico per CodeDeploy, è un file in formato YAML o in formato JSON. Il AppSpec file viene utilizzato per gestire ogni distribuzione come una serie di hook di eventi del ciclo di vita, definiti nel file.

Per informazioni su come creare un file ben formato AppSpec , vedere. CodeDeploy AppSpec riferimento al file

Argomenti

- AppSpec file su una piattaforma di elaborazione Amazon ECS
- AppSpec file su un AWS Lambda piattaforma di calcolo
- AppSpec file su una piattaforma di elaborazione EC2/on-premises
- In che modo l' CodeDeploy agente utilizza il file AppSpec

AppSpec file su una piattaforma di elaborazione Amazon ECS

Se l'applicazione utilizza la piattaforma di calcolo Amazon ECS, il AppSpec file può essere formattato con YAML o JSON. Può anche essere digitato direttamente in un editor nella console. Il AppSpec file viene utilizzato per specificare:

- Il nome del servizio Amazon ECS e il nome e la porta del container utilizzati per indirizzare il traffico verso il nuovo set di attività.
- Le funzioni da utilizzare come test di convalida.

È possibile eseguire le funzioni di convalida Lambda dopo gli eventi del ciclo di vita della distribuzione. Per ulteriori informazioni, consultare AppSpec sezione 'hook' per una distribuzione Amazon ECS, AppSpec struttura dei file per le distribuzioni di Amazon ECS e AppSpec Esempio di file per una distribuzione Amazon ECS.

AppSpec file su un AWS Lambda piattaforma di calcolo

Se l'applicazione utilizza la piattaforma di calcolo AWS Lambda, il AppSpec file può essere formattato con YAML o JSON. Può anche essere digitato direttamente in un editor nella console. Il AppSpec file viene utilizzato per specificare:

- La versione AWS Lambda della funzione da implementare.
- Le funzioni da utilizzare come test di convalida.

È possibile eseguire le funzioni di convalida Lambda dopo gli eventi del ciclo di vita della distribuzione. Per ulteriori informazioni, consulta AppSpec sezione 'hooks' per un AWS Implementazione Lambda.

AppSpec file su una piattaforma di elaborazione EC2/on-premises

Se l'applicazione utilizza la piattaforma di EC2/On-Premises elaborazione, il AppSpec file è sempre disponibile. YAML-formatted Il AppSpec file viene utilizzato per:

- Mappare i file sorgente nella revisione dell'applicazione nelle loro destinazioni sull'istanza.
- Specificare autorizzazioni personalizzate per i file distribuiti.
- Specificare script da eseguire su ciascuna istanza in diverse fasi del processo di distribuzione.

È possibile eseguire script su un'istanza dopo molti dei singoli eventi del ciclo di vita della distribuzione. CodeDeploy esegue solo gli script specificati nel file, ma tali script possono richiamare altri script sull'istanza. Puoi eseguire qualsiasi tipo di script, purché sia supportato dal sistema operativo in esecuzione sulle istanze. Per ulteriori informazioni, consulta AppSpec sezione 'hooks' per una distribuzione EC2/On-Premises.

In che modo l' CodeDeploy agente utilizza il file AppSpec

Durante la distribuzione, l' CodeDeploy agente cerca il nome dell'evento corrente nella sezione hooks del AppSpec file. Se l'evento non viene trovato, l'agente CodeDeploy passa alla fase successiva. Se l'evento viene trovato, l' CodeDeploy agente recupera l'elenco degli script da eseguire. Gli script vengono eseguiti in sequenza, nell'ordine in cui sono mostrati nel file. Lo stato di ogni script viene registrato nel file di registro dell' CodeDeploy agente sull'istanza.

Se uno script viene eseguito con successo, restituisce il codice di uscita 0 (zero).

> **Note**
>
> L' CodeDeploy agente non viene utilizzato in una distribuzione AWS Lambda o Amazon ECS.

Durante l'evento Install, l' CodeDeploy agente utilizza le mappature definite nella sezione files del AppSpec file per determinare quali cartelle o file copiare dalla revisione all'istanza.

Se l' CodeDeploy agente installato sul sistema operativo non corrisponde a quanto elencato nel AppSpec file, la distribuzione non riesce.

Per informazioni sui file di registro CodeDeploy dell'agente, vedereCollaborazione con l' CodeDeploy agente.

Guida introduttiva con CodeDeploy

Argomenti

- Fase 1: Configurazione
- Fase 2: Creare un ruolo di servizio per CodeDeploy
- Passaggio 3: Limita le autorizzazioni CodeDeploy dell'utente
- Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2

Fase 1: Configurazione

Prima di utilizzarlo AWS CodeDeploy per la prima volta, è necessario completare i passaggi di configurazione. I passaggi prevedono la creazione di un AWS account (se non ne hai già uno) e di un utente amministrativo.

In questa guida, l'utente amministrativo è chiamato utente CodeDeploy amministrativo.

Iscriviti per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Esistono due modi per iscriversi AWS. Se desideri il controllo completo sulla configurazione dell'account e sulla posizione in cui sono archiviati i tuoi contenuti, o se hai carichi di lavoro regolamentati come HIPAA o FedRAMP, usa Sign up for (avanzato). AWS Se sei un nuovo utente e desideri utilizzare un nome di accesso che già possiedi per accedere AWS e creare un AWS ambiente con impostazioni preconfigurate preconfigurate in select, usa Sign up for. Regioni AWSAWS

> **Important**
>
> Ti consigliamo vivamente di configurare l'utente CodeDeploy amministrativo come identità della forza lavoro (un utente gestito in IAM Identity Center) con. AWS CLI Molte delle procedure di questa guida presuppongono che tu stia utilizzando il per eseguire le AWS CLI configurazioni.

> **Important**
>
> Se si configura il AWS CLI, è possibile che venga richiesto di specificare una AWS regione. Scegli una delle regioni supportate elencate in Regione e gli endpoint in. Riferimenti generali di AWS

Fase 2: Creare un ruolo di servizio per CodeDeploy

Nel AWS, i ruoli di servizio vengono utilizzati per concedere le autorizzazioni a un AWS servizio in modo che possa accedere alle AWS risorse. Le policy che colleghi al ruolo del servizio determinano le risorse a cui il servizio può accedere e le azioni che può eseguire.

Al ruolo di servizio per cui crei CodeDeploy devono essere concesse le autorizzazioni richieste per la tua piattaforma di elaborazione. Se esegui la distribuzione su più di una piattaforma di elaborazione, crea un ruolo di servizio per ciascuna. Per aggiungere autorizzazioni, allega una o più delle seguenti politiche fornite: AWS

Per le EC2/On-Premises distribuzioni, allega la policy. **`AWSCodeDeployRole`** Fornisce le autorizzazioni necessarie al ruolo del servizio per:

- Leggi i tag sulle tue istanze o identifica le tue istanze Amazon EC2 tramite i nomi dei gruppi di Amazon EC2 Auto Scaling.
- Leggi, crea, aggiorna ed elimina i gruppi, gli hook del ciclo di vita e le politiche di scalabilità di Amazon EC2 Auto Scaling.
- Pubblica informazioni su argomenti di Amazon SNS.
- Recupera informazioni sugli CloudWatch allarmi.
- Leggi e aggiorna Elastic Load Balancing.

> **Note**
>
> Se crei il tuo gruppo Auto Scaling con un modello di avvio, devi aggiungere le seguenti autorizzazioni:
>
> - `ec2:RunInstances`
> - `ec2:CreateTags`
> - `iam:PassRole`

Per ulteriori informazioniFase 2: Creare un ruolo di servizio, consulta Creazione di un modello di lancio per un gruppo Auto Scaling e Supporto dei modelli di avvio nella Guida per l'utente di Amazon EC2 Auto Scaling.

Per le implementazioni di Amazon ECS, se desideri l'accesso completo ai servizi di supporto, allega la policy. **`AWSCodeDeployRoleForECS`** Fornisce le autorizzazioni necessarie al ruolo del servizio per:

- Leggi, aggiorna ed elimina i set di attività di Amazon ECS.
- Aggiorna i gruppi target, gli ascoltatori e le regole di Elastic Load Balancing.
- Invoca funzioni AWS Lambda .
- Accedi ai file di revisione nei bucket Amazon S3.
- Recupera informazioni sugli allarmi. CloudWatch
- Pubblica informazioni su argomenti di Amazon SNS.

Per le implementazioni di Amazon ECS, se desideri un accesso limitato ai servizi di supporto, allega la policy. **`AWSCodeDeployRoleForECSLimited`** Fornisce le autorizzazioni necessarie al ruolo del servizio per:

- Leggi, aggiorna ed elimina i set di attività di Amazon ECS.
- Recupera informazioni sugli CloudWatch allarmi.
- Pubblica informazioni su argomenti di Amazon SNS.

Per le distribuzioni AWS Lambda, se desideri consentire la pubblicazione su Amazon SNS, allega la policy. **`AWSCodeDeployRoleForLambda`** Fornisce le autorizzazioni necessarie al ruolo del servizio per:

- Leggi, aggiorna e richiama AWS Lambda funzioni e alias.
- Accedi ai file di revisione nei bucket Amazon S3.
- Recupera informazioni sugli allarmi. CloudWatch
- Pubblica informazioni su argomenti di Amazon SNS.

Per le implementazioni AWS Lambda, se desideri limitare l'accesso ad Amazon SNS, allega la policy. **`AWSCodeDeployRoleForLambdaLimited`** Fornisce le autorizzazioni necessarie al ruolo del servizio per:

- Leggi, aggiorna e richiama AWS Lambda funzioni e alias.
- Accedi ai file di revisione nei bucket Amazon S3.
- Recupera informazioni sugli allarmi. CloudWatch

Nell'ambito dell'impostazione del ruolo del servizio, si aggiorna anche il suo rapporto di fiducia per specificare gli endpoint a cui si desidera concedere l'accesso.

Puoi creare un ruolo di servizio con la console IAM AWS CLI, o le API IAM.

Argomenti

- Crea un ruolo di servizio (console)
- Creare un ruolo di servizio (CLI)
- Ottieni il ruolo di servizio ARN (console)
- Ottieni il ruolo di servizio ARN (CLI)

Crea un ruolo di servizio (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
3. Scegli il AWS servizio e, in Caso d'uso, dall'elenco a discesa, scegli. CodeDeploy
4. Scegli il tuo caso d'uso:
   - Per le EC2/On-Premises distribuzioni, scegli. CodeDeploy
   - Per le implementazioni AWS Lambda, scegli Lambda. CodeDeploy
   - Per le implementazioni di Amazon ECS, scegli CodeDeploy - ECS.
5. Scegli Next (Successivo).
6. Nella pagina Aggiungi autorizzazioni, viene visualizzata la politica di autorizzazione corretta per il caso d'uso. Scegli Next (Successivo).

7. Nella pagina Nome, revisione e creazione, in Nome ruolo, inserisci un nome per il ruolo di servizio (ad esempio,**`CodeDeployServiceRole`**), quindi scegli Crea ruolo.

   Puoi anche inserire una descrizione per questo ruolo di servizio nella descrizione del ruolo.

8. Se si intende concedere a questo ruolo del servizio l'autorizzazione per accedere a tutti gli endpoint attualmente supportati, la procedura è terminata.

   Per limitare l'accesso di questo ruolo di servizio ad alcuni endpoint, continua con i passaggi rimanenti di questa procedura.

9. Nell'elenco dei ruoli, cerca e scegli il ruolo che hai appena creato (`CodeDeployServiceRole`).
10. Seleziona la scheda Relazioni di attendibilità
11. Seleziona Modifica policy di attendibilità.

    Dovrebbero essere visualizzate le seguenti policy, che offrono al ruolo del servizio l'autorizzazione per accedere a tutti gli endpoint supportati:

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

    Per concedere al ruolo di servizio l'accesso solo ad alcuni endpoint supportati, sostituisci il contenuto della casella di testo della politica di fiducia con la seguente politica. Rimuovi le righe relative agli endpoint a cui vuoi impedire l'accesso, quindi scegli Aggiorna policy.

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

Per ulteriori informazioni sulla creazione di ruoli di servizio, consulta Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'utente IAM.

Creare un ruolo di servizio (CLI)

1. Nel computer di sviluppo creare un file di testo denominato, ad esempio `CodeDeployDemo-Trust.json`. Questo file viene utilizzato per consentire a CodeDeploy di lavorare per conto dell'utente.

   Esegui una delle seguenti operazioni:

   - Per concedere l'accesso a tutte le AWS regioni supportate, salva il seguente contenuto nel file:

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

   - Per concedere l'accesso solo ad alcune regioni supportate, digita il seguente contenuto nel file e rimuovi le righe relative alle regioni a cui desideri escludere l'accesso:

```
{
    "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**
>
> Non inserire la virgola dopo l'ultimo endpoint dell'elenco.

2. Dalla stessa directory, chiama il comando create-role per creare un ruolo del servizio denominato **`CodeDeployServiceRole`** in base alle informazioni del file di testo appena creato:

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

> **Important**
>
> Assicurarsi di includere `file://` prima del nome del file. Questo è obbligatorio in questo comando.

Nell'output del comando prendere nota del valore della voce `Arn` sotto l'oggetto `Role`. Servirà successivamente al momento della creazione dei gruppi di distribuzione. Se si dimentica il valore, attenersi alle istruzioni in Ottieni il ruolo di servizio ARN (CLI).

3. La policy gestita utilizzata dipende dalla piattaforma di elaborazione.

   - Se la distribuzione è su una piattaforma di EC2/On-Premises elaborazione:

     Chiama il attach-role-policy comando per assegnare al ruolo di servizio denominato **`CodeDeployServiceRole`** le autorizzazioni in base alla policy gestita IAM denominata. **`AWSCodeDeployRole`** Ad esempio:

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

   - Se la distribuzione è su una piattaforma di elaborazione AWS Lambda:

     Chiama il attach-role-policy comando per assegnare al ruolo di servizio denominato **`CodeDeployServiceRole`** le autorizzazioni in base alla policy gestita IAM denominata o. **`AWSCodeDeployRoleForLambda`** **`AWSCodeDeployRoleForLambdaLimited`** Ad esempio:

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

   - Se la distribuzione è su una piattaforma di calcolo Amazon ECS:

     Chiama il attach-role-policy comando per assegnare al ruolo di servizio denominato **`CodeDeployServiceRole`** le autorizzazioni in base alla policy gestita IAM denominata or. **`AWSCodeDeployRoleForECS`** **`AWSCodeDeployRoleForECSLimited`** Ad esempio:

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

Per ulteriori informazioni sulla creazione di ruoli di servizio, consulta Creating a role for an AWS service nella IAM User Guide.

Ottieni il ruolo di servizio ARN (console)

Per utilizzare la console IAM per ottenere l'ARN del ruolo di servizio:

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Nella casella Filter (Filtro) digitare **`CodeDeployServiceRole`** e premere Invio.
4. Scegli CodeDeployServiceRole.
5. Prendere nota del valore del campo Role ARN (ARN ruolo).

Ottieni il ruolo di servizio ARN (CLI)

Per utilizzare il AWS CLI per ottenere l'ARN del ruolo di servizio, chiama il get-role comando sul ruolo di servizio denominato: **`CodeDeployServiceRole`**

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

Il valore restituito è l'ARN del ruolo del servizio.

Passaggio 3: Limita le autorizzazioni CodeDeploy dell'utente

Per motivi di sicurezza, ti consigliamo di limitare le autorizzazioni dell'utente amministrativo che hai creato solo Fase 1: Configurazione a quelle necessarie per creare e gestire le distribuzioni in. CodeDeploy

Utilizza la seguente serie di procedure per limitare le autorizzazioni dell'utente CodeDeploy amministrativo.

Prima di iniziare

- Assicurati di aver creato un utente CodeDeploy amministrativo in IAM Identity Center seguendo le istruzioni riportate inFase 1: Configurazione.

Per creare un set di autorizzazioni

Assegnerai questo set di autorizzazioni all'utente CodeDeploy amministrativo in un secondo momento.

1. Accedi a Console di gestione AWS e apri la Centro identità AWS IAM console all'indirizzo https://console.aws.amazon.com/singlesignon/.
2. Nel riquadro di navigazione, scegli Set di autorizzazioni, quindi scegli Crea set di autorizzazioni.
3. Scegli Set di autorizzazioni personalizzato.
4. Scegli Avanti.
5. Scegli Inline policy.
6. Rimuovi il codice di esempio.
7. Aggiungi il seguente codice di policy:

   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"
        }
    ]
}
```

In questo criterio, sostituiscilo *arn:aws:iam::account-ID:role/CodeDeployServiceRole* con il valore ARN del ruolo di CodeDeploy servizio in cui hai creato. Fase 2: Creare un ruolo di servizio per CodeDeploy Puoi trovare il valore ARN nella pagina dei dettagli del ruolo di servizio nella console IAM.

La politica precedente consente di distribuire un'applicazione su una piattaforma di calcolo AWS Lambda, una piattaforma di elaborazione e una EC2/On-Premises piattaforma di calcolo Amazon ECS.

Puoi utilizzare i CloudFormation modelli forniti in questa documentazione per avviare istanze Amazon EC2 compatibili con. CodeDeploy Per utilizzare i CloudFormation modelli per creare applicazioni, gruppi di distribuzione o configurazioni di distribuzione, devi fornire l'accesso ai AWS servizi CloudFormation e alle azioni da cui CloudFormation dipendono aggiungendo l'`cloudformation:*`autorizzazione alla politica di autorizzazione dell'utente CodeDeploy amministrativo, come segue:

JSON

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

8. Scegli Avanti.
9. Nel nome del set di autorizzazioni, inserisci:

```
CodeDeployUserPermissionSet
```

10. Scegli Avanti.
11. Nella pagina Rivedi e crea, esamina le informazioni e scegli Crea.

Per assegnare il set di autorizzazioni all'utente CodeDeploy amministrativo

1. Nel riquadro di navigazione, scegli Account AWS, quindi seleziona la casella di controllo accanto a Account AWS quella a cui hai attualmente effettuato l'accesso.
2. Scegli il pulsante Assegna utenti o gruppi.
3. Scegli la scheda Users (Utenti);
4. Seleziona la casella di controllo accanto all'utente CodeDeploy amministrativo.
5. Scegli Avanti.
6. Seleziona la casella di controllo accanto a`CodeDeployUserPermissionSet`.
7. Scegli Avanti.
8. Controlla le informazioni e scegli Invia.

Ora hai assegnato l'utente CodeDeploy amministrativo e `CodeDeployUserPermissionSet` il tuo utente Account AWS, associandoli insieme.

Per disconnetterti e accedere nuovamente come utente CodeDeploy amministrativo

1. Prima di disconnetterti, assicurati di avere l'URL del portale di AWS accesso e il nome utente e la password monouso per l'utente CodeDeploy amministrativo.

   > **Note**
   >
   > Se non disponi di queste informazioni, vai alla pagina dei dettagli utente CodeDeploy amministrativi in IAM Identity Center, scegli Reimposta password, Genera una password monouso [...] e Reimposta nuovamente la password per visualizzare le informazioni sullo schermo.

2. Esci da AWS.
3. Incolla l'URL del portale di AWS accesso nella barra degli indirizzi del browser.
4. Accedi come utente CodeDeploy amministrativo.

   Sullo schermo viene visualizzata una Account AWScasella.

5. Scegli Account AWS, quindi scegli il nome Account AWS a cui hai assegnato l'utente CodeDeploy amministrativo e il set di autorizzazioni.
6. Accanto a**`CodeDeployUserPermissionSet`**, scegli Console di gestione.

   Console di gestione AWS Appare. Ora hai effettuato l'accesso come utente CodeDeploy amministrativo con le autorizzazioni limitate. Ora puoi eseguire CodeDeploy-related operazioni, e solo CodeDeploy-related operazioni, come questo utente.

Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2

> **Note**
>
> Se utilizzi la piattaforma di calcolo Amazon ECS o AWS Lambda, salta questo passaggio.

Le tue istanze Amazon EC2 necessitano dell'autorizzazione per accedere ai bucket o ai GitHub repository Amazon S3 in cui sono archiviate le applicazioni. Per avviare istanze Amazon EC2 compatibili con CodeDeploy, devi creare un ruolo IAM aggiuntivo, un profilo di istanza. Queste istruzioni mostrano come creare un profilo di istanza IAM da collegare alle istanze Amazon EC2. Questo ruolo consente all' CodeDeploy agente di accedere ai bucket o ai GitHub repository Amazon S3 in cui sono archiviate le tue applicazioni.

Puoi creare un profilo di istanza IAM con AWS CLI, la console IAM o le API IAM.

> **Note**
>
> È possibile collegare un profilo dell'istanza IAM a un'istanza Amazon EC2 all'avvio o a un'istanza avviata in precedenza. Per ulteriori informazioni, consulta Profili di istanza.

Argomenti

- Crea un profilo di istanza IAM per le tue istanze Amazon EC2 (CLI)
- Crea un profilo di istanza IAM per le tue istanze Amazon EC2 (console)

Crea un profilo di istanza IAM per le tue istanze Amazon EC2 (CLI)

In queste fasi, presumiamo che tu abbia già seguito le istruzioni per le prime tre fasi di Guida introduttiva con CodeDeploy.

1. Nella macchina di sviluppo, creare un file di testo denominato `CodeDeployDemo-EC2-Trust.json`. Incolla il seguente contenuto, che consente ad Amazon EC2 di lavorare per tuo conto:

   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. Nella stessa directory, crea un file di testo denominato `CodeDeployDemo-EC2-Permissions.json`. Incolla il contenuto seguente:

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

Note

Ti consigliamo di limitare questa politica solo ai bucket Amazon S3 a cui devono accedere le tue istanze Amazon EC2. Assicurati di consentire l'accesso ai bucket Amazon S3 che contengono l'agente. CodeDeploy In caso contrario, potrebbe verificarsi un errore quando l' CodeDeploy agente viene installato o aggiornato sulle istanze.
Per concedere al profilo dell'istanza IAM l'accesso solo ad alcuni bucket del kit di CodeDeploy risorse in Amazon S3, utilizza la seguente policy, ma rimuovi le righe relative ai bucket a cui desideri impedire l'accesso:

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

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

> **Note**
>
> Se desideri utilizzare l'autorizzazione IAM o gli endpoint Amazon Virtual Private Cloud (VPC) con CodeDeploy, dovrai aggiungere altre autorizzazioni. Per ulteriori informazioni, consulta Utilizzo CodeDeploy con Amazon Virtual Private Cloud.

3. Dalla stessa directory, chiama il create-role comando per creare un ruolo IAM denominato**`CodeDeployDemo-EC2-Instance-Profile`**, in base alle informazioni contenute nel primo file:

> **Important**
>
> Assicurarsi di includere `file://` prima del nome del file. Questo è obbligatorio in questo comando.

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

4. Dalla stessa directory chiama il comando put-role-policy per concedere a un ruolo denominato **`CodeDeployDemo-EC2-Instance-Profile`** le autorizzazioni in base alle informazioni del secondo file:

> **Important**
>
> Assicurarsi di includere `file://` prima del nome del file. Questo è obbligatorio in questo comando.

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

5. Chiamalo attach-role-policy per concedere al ruolo le autorizzazioni di Amazon EC2 Systems Manager in modo che SSM possa installare l'agente. CodeDeploy Questa politica non è necessaria se prevedi di installare l'agente dal bucket pubblico di Amazon S3 con la riga di comando. Ulteriori informazioni sull'installazione dell'agente CodeDeploy .

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

6. Chiama il create-instance-profile comando seguito dal add-role-to-instance-profile comando per creare un profilo di istanza IAM denominato. **`CodeDeployDemo-EC2-Instance-Profile`** Il profilo dell'istanza consente ad Amazon EC2 di passare il ruolo IAM denominato **`CodeDeployDemo-EC2-Instance-Profile`** a un'istanza Amazon EC2 al primo avvio dell'istanza:

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

   Se hai bisogno di ottenere il nome del profilo dell'istanza IAM, consulta list-instance-profiles-for-role nella sezione IAM del Reference.AWS CLI

Ora hai creato un profilo di istanza IAM da collegare alle tue istanze Amazon EC2. Per ulteriori informazioni, consulta Ruoli IAM per Amazon EC2 nella Guida per l'utente di Amazon EC2.

Crea un profilo di istanza IAM per le tue istanze Amazon EC2 (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nella console IAM, nel riquadro di navigazione, scegli Policies, quindi scegli Crea policy.
3. Nella pagina Specifica autorizzazioni, seleziona JSON.
4. Rimuovi il `JSON` codice di esempio.
5. Incolla il codice seguente:

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

> **Note**
>
> Ti consigliamo di limitare questa politica solo ai bucket Amazon S3 a cui devono accedere le tue istanze Amazon EC2. Assicurati di consentire l'accesso ai bucket Amazon S3 che contengono l'agente. CodeDeploy In caso contrario, potrebbe verificarsi un errore quando l' CodeDeploy agente viene installato o aggiornato sulle istanze. Per concedere al profilo dell'istanza IAM l'accesso solo ad alcuni bucket del kit di CodeDeploy risorse in Amazon S3, utilizza la seguente policy, ma rimuovi le righe relative ai bucket a cui desideri impedire l'accesso:
>
> ```
> {
>   "Version": "2012-10-17",
>   "Statement": [
>     {
>       "Effect": "Allow",
>       "Action": [
> ```

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

> **Note**
>
> Se desideri utilizzare l'autorizzazione IAM o gli endpoint Amazon Virtual Private Cloud (VPC) con CodeDeploy, dovrai aggiungere altre autorizzazioni. Per ulteriori informazioni, consulta Utilizzo CodeDeploy con Amazon Virtual Private Cloud.

6. Scegli Next (Successivo).
7. Nella pagina Rivedi e crea, nella casella Nome della politica, digita**`CodeDeployDemo-EC2-Permissions`**.
8. (Facoltativo) In Description (Descrizione) inserire una descrizione per la policy.
9. Scegli Crea policy.
10. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
11. In Caso d'uso, scegli il caso d'uso EC2.
12. Scegli Next (Successivo).
13. Nell'elenco delle politiche, seleziona la casella di controllo accanto alla politica che hai appena creato (CodeDeployDemo-EC2-Permissions). Se necessario, utilizzare la casella di ricerca per trovare la policy.
14. Per utilizzare Systems Manager per installare o configurare l' CodeDeploy agente, selezionare la casella di controllo accanto a AmazonSSMManagedInstanceCore. Questa policy AWS gestita consente a un'istanza di utilizzare le funzionalità principali del servizio Systems Manager. Se necessario, utilizzare la casella di ricerca per trovare la policy. Questa politica non è necessaria se prevedi di installare l'agente dal bucket pubblico di Amazon S3 con la riga di comando. Ulteriori informazioni sull'installazione dell'agente CodeDeploy.
15. Scegli Next (Successivo).
16. Nella pagina Nome, revisione e creazione, in Nome ruolo, inserisci un nome per il ruolo di servizio (ad esempio,**`CodeDeployDemo-EC2-Instance-Profile`**), quindi scegli Crea ruolo.

    Puoi anche inserire una descrizione per questo ruolo di servizio nella descrizione del ruolo.

Ora hai creato un profilo di istanza IAM da collegare alle tue istanze Amazon EC2. Per ulteriori informazioni, consulta Ruoli IAM per Amazon EC2 nella Guida per l'utente di Amazon EC2.

Integrazioni di prodotti e servizi con CodeDeploy

Per impostazione predefinita, CodeDeploy si integra con una serie di AWS servizi e prodotti e servizi dei partner. Le seguenti informazioni possono aiutarti CodeDeploy a configurare l'integrazione con i prodotti e i servizi che utilizzi.

- Integrazione con altri AWS services
- Integrazione con prodotti e servizi dei partner
- Esempi di integrazione dalla community

Integrazione con altri AWS services

CodeDeploy è integrato con i seguenti AWS servizi:

Amazon CloudWatch	Amazon CloudWatch è un servizio di monitoraggio per le risorse AWS cloud e le applicazioni su cui esegui AWS. Puoi usare Amazon CloudWatch per raccogliere e tracciare metriche, raccogliere e monitorare file di registro e impostare allarmi. CodeDeploy supporta i seguenti strumenti: CloudWatch • CloudWatch allarmi per monitorare le implementazioni e interromperle quando le metriche di monitoraggio specificate superano o scendono al di sotto delle soglie specificate in una regola di allarme. CloudWatch Per utilizzare il monitoraggio degli allarmi, è necessario innanzitutto impostare un allarme e quindi aggiungerlo all'applicazione o CodeDeploy al gruppo di distribuzione in cui le distribuzioni devono interrompersi quando viene attivato l'allarme. CloudWatch Ulteriori informazioni:

- Creazione di registri CloudWatch e allarmi

- Amazon CloudWatch Events per rilevare e reagire ai cambiamenti nello stato di un'istanza o di una distribuzione nelle tue CodeDeploy operazioni. Quindi, in base alle regole che crei, CloudWatch Events richiama una o più azioni mirate quando una distribuzione o un'istanza entra nello stato specificato in una regola.

  Ulteriori informazioni:

  - Monitoraggio delle distribuzioni con Amazon Events CloudWatch

- Amazon CloudWatch Logs per monitorare i tre tipi di log creati dall' CodeDeploy agente senza dover accedere alle istanze una alla volta.

  Ulteriori informazioni:

  - Visualizza i CodeDeploy log nella console Logs CloudWatch

Amazon EC2 Auto Scaling	CodeDeploy supporta Amazon EC2 Auto Scaling. Questo AWS servizio può avviare automaticamente istanze Amazon EC2 in base a criteri specificati, ad esempio: • Superamento dei limiti per l'utilizzo della CPU specificati. • Letture o scritture su disco • Il traffico di rete in entrata o in uscita per un intervallo di tempo specificato. Puoi scalare orizzontalmente un gruppo di istanze Amazon EC2 ogni volta che ne hai bisogno e poi utilizzarle CodeDeploy per distribuire automaticamente le revisioni delle applicazioni. Amazon EC2 Auto Scaling interrompe le istanze Amazon EC2 quando non sono più necessarie. Ulteriori informazioni: • Integrazione CodeDeploy con Amazon EC2 Auto Scaling • Tutorial: CodeDeploy Da utilizzare per distribuire un'applicazione in un gruppo di Auto Scaling • Sotto il cofano: CodeDeploy e l'integrazione con Auto Scaling

Amazon Elastic Container Service	Puoi utilizzarla CodeDeploy per distribuire un'applicazione containerizzata Amazon ECS come set di attività. CodeDeploy esegue una blue/green distribuzione installando una versione aggiornata dell'applicazione come nuovo set di attività sostitutivo. CodeDeploy reindirizza il traffico di produzione dal set di attività dell'applicazione originale al set di attività sostitutivo. Il set di attività originale viene terminato una volta completata l'implementazione. Per ulteriori informazioni su Amazon ECS, consulta Amazon Elastic Container Service. È possibile gestire il modo in cui il traffico viene spostato al set di attività aggiornato durante una distribuzione scegliendo una configurazione canary, lineare o all-at-once. Per ulteriori informazioni sulle distribuzioni di Amazon ECS, consulta Distribuzioni su una piattaforma di calcolo Amazon ECS.

AWS CloudTrail	CodeDeploy è integrato con. AWS CloudTrail Questo servizio acquisisce le chiamate API effettuate da o per conto del CodeDeploy y tuo AWS account e invia i file di log a un bucket Amazon S3 da te specificato. CloudTrai lacquisisce le chiamate API dalla CodeDeplo y console, dai CodeDeploy comandi tramite o direttamente AWS CLI dalle API. CodeDeplo y Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare: • A quale richiesta è stata fatta. CodeDeploy • L'indirizzo IP di origine dal quale è stata effettuata la richiesta. • Chi ha effettuato la richiesta. • Quando è stata effettuata. Ulteriori informazioni: • Monitoring Deployments

AWS Cloud9	AWS Cloud9è un ambiente di sviluppo integrato (IDE) online basato sul cloud che puoi utilizzare per scrivere, eseguire, eseguire il debug e distribuire codice utilizzando solo un browser da una macchina connessa a Internet. AWS Cloud9 include un editor di codice, un debugger, un terminale e strumenti essenziali, come Git e AWS CLI Git. • È possibile utilizzare l' AWS Cloud9 IDE per eseguire, eseguire il debug e creare codice contenuto in un repository. GitHub Puoi visualizzare, modificare e salvare il codice utilizzando la finestra Environment (Ambiente) e le schede dell'editor nell'IDE. Quando sei pronto, puoi usare Git nella sessione del AWS Cloud9 terminale per inviare modifiche al codice nel tuo GitHub repository e poi usarlo AWS CodeDeploy per distribuire gli aggiornamenti. Per ulteriori informazioni sull'utilizzo di AWS Cloud9 with GitHub, consulta l'GitHub esempio per. AWS Cloud9 • È possibile utilizzare l' AWS Cloud9 IDE per aggiornare una AWS Lambda funzione. Puoi quindi AWS CodeDeploy utilizzarlo per creare una distribuzione che sposti il traffico verso la nuova versione della tua AWS Lambda funzione. Per ulteriori informazioni, vedere Utilizzo AWS Lambda delle funzioni nell' AWS Cloud9 Integrated Development Environment (IDE). Per ulteriori informazioni su AWS Cloud9, consulta Cos'è AWS Cloud9 e Guida introduttiva AWS Cloud9.

AWS CodePipeline	AWS CodePipeline è un servizio di distribuzione continua che può essere utilizzato per modellare, visualizzare e automatizzare le fasi necessarie al rilascio di software in un processo di distribuzione continua. Puoi utilizzare AWS CodePipeline per definire un tuo processo di rilascio in modo che il servizio compili, testi e distribuisca il codice ogni volta che questo viene modificato. Ad esempio, potresti disporre di tre gruppi di distribuzione per un'applicazione: Beta, Gamma e Prod. Puoi impostare una pipeline in modo che ogni volta che viene modificato il codice sorgente, gli aggiornamenti vengano distribuiti a ciascun gruppo di distribuzione singolarmente. È possibile configurare AWS CodePipeline per l'utilizzo CodeDeploy per la distribuzione: • Codifica per istanze Amazon EC2, istanze locali o entrambe. • Versioni con funzione AWS Lambda serverless. È possibile creare l' CodeDeploy applicazione, la distribuzione e il gruppo di distribuzione da utilizzare in un'azione di distribuzione in una fase prima di creare la pipeline o nella procedura guidata Create Pipeline. Ulteriori informazioni: • AWS guida DevOps introduttiva: scopri come utilizzare CodePipeline per distribuire e CodeDeploy distribuire continuamente il

	codice sorgente nei CodeCommit repository sulle istanze Amazon EC2. • Semplice procedura dettagliata della pipeline (Amazon S3 Bucket) • Semplice procedura dettagliata della pipeline (Repository) CodeCommit • Four-stagetutorial sulla pipeline
AWS Modello di applicazione serverless	AWS Serverless Application Model (AWS SAM) è un modello per definire applicazioni serverless. Si estende CloudFormation per fornire un modo semplificato di definire AWS Lambda funzioni, API Amazon API Gateway e tabelle Amazon DynamoDB richieste da un'applicazione serverless. Se utilizzi già AWS SAM, puoi aggiungere preferenze di distribuzione per iniziare CodeDeploy a utilizzare per gestire il modo in cui il traffico viene spostato durante la distribuzione di un'applicazione AWS Lambda. Per ulteriori informazioni, consulta il modello di applicazione AWS Serverless.

Elastic Load Balancing	CodeDeploy supporta Elastic Load Balancing , un servizio che distribuisce il traffico delle applicazioni in entrata su più istanze Amazon EC2. Per quanto riguarda CodeDeploy le implement azioni, i sistemi di bilanciamento del carico impediscono inoltre che il traffico venga indirizzato verso istanze quando non sono pronte, sono attualmente in fase di implement azione o non sono più necessarie come parte di un ambiente. Ulteriori informazioni: • Integrating CodeDeploy with Elastic Load Balancing

Argomenti

- Integrazione CodeDeploy con Amazon EC2 Auto Scaling
- Integrazione CodeDeploy con Elastic Load Balancing

Integrazione CodeDeploy con Amazon EC2 Auto Scaling

CodeDeploy supporta Amazon EC2 Auto Scaling AWS , un servizio che avvia automaticamente le istanze Amazon EC2 in base alle condizioni da te definite. Queste condizioni possono includere limiti superati in un intervallo di tempo specificato per l'utilizzo della CPU, le letture o le scritture del disco o il traffico di rete in entrata o in uscita. Amazon EC2 Auto Scaling interrompe le istanze quando non sono più necessarie. Per maggiori informazioni, consulta What is Amazon EC2 Auto Scaling? nella Guida per l'utente di Amazon EC2 Auto Scaling.

Quando vengono lanciate nuove istanze Amazon EC2 come parte di un CodeDeploy gruppo Amazon EC2 Auto Scaling, puoi distribuire automaticamente le tue revisioni alle nuove istanze. Puoi anche coordinare le distribuzioni con istanze Amazon EC2 Auto Scaling registrate CodeDeploy con i sistemi di bilanciamento del carico Elastic Load Balancing. Per ulteriori informazioni, consultare Integrating

CodeDeploy with Elastic Load Balancing e Configura un sistema di bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2.

> **Note**
>
> Potresti riscontrare problemi se associ più gruppi di distribuzione a un singolo gruppo Amazon EC2 Auto Scaling. Se, ad esempio, una distribuzione ha esito negativo, l'istanza inizierà l'arresto, ma le altre distribuzioni che erano in esecuzione possono richiedere un'ora di tempo per scadere. Per ulteriori informazioni, consulta Evita di associare più gruppi di distribuzione a un singolo gruppo Amazon EC2 Auto Scaling Under the hood: CodeDeploy and Amazon EC2 Auto Scaling integration.

Argomenti

- Distribuzione di CodeDeploy applicazioni su gruppi Amazon EC2 Auto Scaling
- Abilitazione delle implementazioni di terminazione durante gli eventi scale-in di Auto Scaling
- Come funziona Amazon EC2 Auto Scaling con CodeDeploy
- Utilizzo di un'AMI personalizzata con CodeDeploy Amazon EC2 Auto Scaling

Distribuzione di CodeDeploy applicazioni su gruppi Amazon EC2 Auto Scaling

Per distribuire una revisione CodeDeploy dell'applicazione in un gruppo Amazon EC2 Auto Scaling:

1. Crea o individua un profilo di istanza IAM che consenta al gruppo Amazon EC2 Auto Scaling di lavorare con Amazon S3. Per ulteriori informazioni, consulta Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2.

   > **Note**
   >
   > Puoi anche utilizzare CodeDeploy per distribuire le revisioni dai GitHub repository ai gruppi Amazon EC2 Auto Scaling. Sebbene le istanze Amazon EC2 richiedano ancora un profilo di istanza IAM, il profilo non necessita di autorizzazioni aggiuntive per la distribuzione da un repository. GitHub

2. Crea o usa un gruppo Amazon EC2 Auto Scaling, specificando il profilo dell'istanza IAM nella configurazione o nel modello di lancio. Per ulteriori informazioni, consulta il ruolo IAM per le applicazioni eseguite su istanze Amazon EC2.

3. Crea o individua un ruolo di servizio che CodeDeploy consenta di creare un gruppo di distribuzione che contenga il gruppo Amazon EC2 Auto Scaling.
4. Crea un gruppo di CodeDeploy distribuzione specificando il nome del gruppo Amazon EC2 Auto Scaling, il ruolo del servizio e alcune altre opzioni. Per ulteriori informazioni, consulta Crea un gruppo di distribuzione per una distribuzione sul posto (console) o Crea un gruppo di distribuzione per una distribuzione sul posto (console).
5. Usalo CodeDeploy per distribuire la tua revisione nel gruppo di distribuzione che contiene il gruppo Amazon EC2 Auto Scaling.

Per ulteriori informazioni, consulta Tutorial: CodeDeploy Da utilizzare per distribuire un'applicazione in un gruppo di Auto Scaling.

Abilitazione delle implementazioni di terminazione durante gli eventi scale-in di Auto Scaling

Una distribuzione con interruzione è un tipo di CodeDeploy distribuzione che viene attivata automaticamente quando si verifica un evento di scalabilità in Auto Scaling. CodeDeploy esegue la distribuzione di terminazione subito prima che il servizio Auto Scaling termini l'istanza. Durante una distribuzione al termine, CodeDeploy non distribuisce nulla. Genera invece eventi del ciclo di vita, che è possibile collegare ai propri script per abilitare funzionalità di spegnimento personalizzate. Ad esempio, è possibile collegare l'evento del `ApplicationStop` ciclo di vita a uno script che chiude correttamente l'applicazione prima che l'istanza venga terminata.

Per un elenco degli eventi del ciclo di vita generati durante una distribuzione di terminazione, consulta CodeDeploy . Disponibilità di Lifecycle Event Hook

Se la distribuzione di terminazione fallisce per qualsiasi motivo, CodeDeploy consentirà alla chiusura dell'istanza di procedere. Ciò significa che l'istanza verrà chiusa anche se CodeDeploy non è stato eseguito l'intero set (o nessuno) degli eventi del ciclo di vita fino al completamento.

Se non abiliti le distribuzioni di terminazione, il servizio Auto Scaling interromperà comunque le istanze Amazon EC2 quando si verifica un evento di scalabilità in, ma non genererà eventi del ciclo di vita. CodeDeploy

> **Note**
>
> Indipendentemente dal fatto che tu abiliti o meno le implementazioni di terminazione, se il servizio Auto Scaling termina un'istanza Amazon EC2 mentre è in corso un'implementazione,

potrebbe verificarsi CodeDeploy una situazione di gara tra gli eventi del ciclo di vita generati da Auto Scaling e i servizi. CodeDeploy Ad esempio, l'evento del `Terminating` ciclo di vita (generato dal servizio Auto Scaling) potrebbe sovrascrivere `ApplicationStart` l'evento (generato dalla distribuzione). CodeDeploy In questo scenario, è possibile che si verifichi un errore con la chiusura dell'istanza Amazon EC2 o con la distribuzione. CodeDeploy

Per consentire l'esecuzione di CodeDeploy distribuzioni di terminazione

- Seleziona la casella di controllo Aggiungi un termination hook ai gruppi di Auto Scaling durante la creazione o l'aggiornamento del gruppo di distribuzione. Per istruzioni, consulta Crea un gruppo di distribuzione per una distribuzione sul posto (console) o. Crea un gruppo di distribuzione per una distribuzione (console) EC2/On-Premises blue/green

  L'attivazione di questa casella di controllo comporta CodeDeploy l'installazione di un hook del ciclo di vita di Auto Scaling nei gruppi Auto Scaling specificati quando si crea o si aggiorna il gruppo di distribuzione. CodeDeploy Questo hook è chiamato termination hook e consente le distribuzioni di terminazione.

Dopo l'installazione dell'hook di terminazione, si svolge un evento scale-in (terminazione) come segue:

1. Il servizio Auto Scaling (o semplicemente Auto Scaling) determina che deve verificarsi un evento di scale-in e contatta il servizio EC2 per terminare un'istanza EC2.
2. Il servizio EC2 inizia a terminare l'istanza EC2. L'istanza passa allo `Terminating` stato e quindi allo stato. `Terminating:Wait`
3. Durante`Terminating:Wait`, Auto Scaling esegue tutti gli hook del ciclo di vita collegati al gruppo Auto Scaling, incluso l'hook di terminazione installato da. CodeDeploy
4. L'hook di terminazione invia una notifica alla coda Amazon SQS da cui viene effettuato il polling. CodeDeploy
5. Dopo aver ricevuto la notifica, CodeDeploy analizza il messaggio, esegue alcune convalide ed esegue una distribuzione di terminazione.
6. Mentre la distribuzione di terminazione è in esecuzione, CodeDeploy invia battiti cardiaci ogni cinque minuti ad Auto Scaling per informarlo che l'istanza è ancora in fase di elaborazione.
7. Finora, l'istanza EC2 è ancora nello `Terminating:Wait` stato (o forse nello `Warmed:Pending:Wait` stato, se hai abilitato i pool caldi di gruppo Auto Scaling).

8. Al termine della distribuzione, CodeDeploy indica il processo di terminazione di Auto Scaling `CONTINUE` to the EC2, indipendentemente dal fatto che l'implementazione di terminazione sia riuscita o meno.

Come funziona Amazon EC2 Auto Scaling con CodeDeploy

Quando crei o aggiorni un gruppo di CodeDeploy distribuzione per includere un gruppo Auto Scaling, CodeDeploy accede al gruppo Auto Scaling utilizzando il ruolo di CodeDeploy servizio, quindi installa gli hook del ciclo di vita di Auto Scaling nei gruppi Auto Scaling.

> **Note**
>
> Gli hook del ciclo di vita di Auto Scaling sono diversi dagli eventi del ciclo di vita (chiamati anche hook degli eventi del ciclo di vita) generati e descritti nella presente guida. CodeDeploy AppSpec sezione 'hooks'

Gli hook del ciclo di vita di Auto Scaling che vengono installati sono: CodeDeploy

- Un hook di lancio: questo hook notifica CodeDeploy che è in corso un evento di scalabilità orizzontale di Auto Scaling e CodeDeploy che è necessario avviare una distribuzione di lancio.

  Durante una distribuzione di lancio,: CodeDeploy

  - Implementa una revisione dell'applicazione nell'istanza scalabile.
  - Genera eventi del ciclo di vita per indicare lo stato di avanzamento della distribuzione. È possibile collegare questi eventi del ciclo di vita ai propri script per abilitare funzionalità di avvio personalizzate. Per ulteriori informazioni, consulta la tabella in. Disponibilità di Lifecycle Event Hook

  Il launch hook e la distribuzione di avvio associata sono sempre abilitati e non possono essere disattivati.

- Un hook di terminazione: questo hook opzionale notifica CodeDeploy che è in corso un evento di scale-in Auto Scaling e CodeDeploy che è necessario avviare una distribuzione di terminazione.

  Durante una distribuzione con interruzione, CodeDeploy genera eventi del ciclo di vita per indicare lo stato di avanzamento della chiusura dell'istanza. Per ulteriori informazioni, consulta Abilitazione delle implementazioni di terminazione durante gli eventi scale-in di Auto Scaling.

Argomenti

- Dopo aver CodeDeploy installato i lifecycle hook, come vengono utilizzati?
- Come vengono CodeDeploy denominati i gruppi Amazon EC2 Auto Scaling
- Ordine di esecuzione degli eventi hook del ciclo di vita personalizzati
- Scale-out eventi durante una distribuzione
- Scale-in eventi durante una distribuzione
- Ordine degli eventi in AWS CloudFormation script cfn-init

Dopo aver CodeDeploy installato i lifecycle hook, come vengono utilizzati?

Dopo aver installato gli hook del ciclo di vita di avvio e terminazione, vengono utilizzati rispettivamente durante gli eventi di scalabilità orizzontale e di CodeDeploy scalabilità in gruppo Auto Scaling.

Un evento di scale-out (lancio) si svolge come segue:

1. Il servizio Auto Scaling (o semplicemente Auto Scaling) determina la necessità di un evento di scalabilità orizzontale e contatta il servizio EC2 per lanciare una nuova istanza EC2.
2. Il servizio EC2 lancia una nuova istanza EC2. L'istanza passa allo `Pending` stato e quindi allo stato. `Pending:Wait`
3. Durante`Pending:Wait`, Auto Scaling esegue tutti gli hook del ciclo di vita collegati al gruppo Auto Scaling, incluso l'hook di avvio installato da. CodeDeploy
4. L'hook di avvio invia una notifica alla coda di Amazon SQS da cui viene effettuato il polling. CodeDeploy
5. Dopo aver ricevuto la notifica, CodeDeploy analizza il messaggio, esegue alcune convalide e avvia una distribuzione di avvio.
6. Mentre la distribuzione di lancio è in esecuzione, CodeDeploy invia battiti cardiaci ogni cinque minuti ad Auto Scaling per informarlo che l'istanza è ancora in fase di elaborazione.
7. Finora, l'istanza EC2 è ancora nello stato. `Pending:Wait`
8. Al termine dell'implementazione, CodeDeploy indica Auto Scaling su uno dei `CONTINUE` due `ABANDON` o sul processo di avvio di EC2, a seconda che l'implementazione sia riuscita o meno.
   - Se lo CodeDeploy indica`CONTINUE`, Auto Scaling continua il processo di avvio, aspettando il completamento degli altri hook oppure inserendo l'istanza nello stato `Pending:Proceed` e quindi nello stato. `InService`

- Se lo CodeDeploy indica**`ABANDON`**, Auto Scaling termina l'istanza EC2 e riavvia la procedura di avvio se necessario per soddisfare il numero di istanze desiderato, come definito nell'impostazione Auto Scaling Desired Capacity.

Un evento di scale-in (terminazione) si svolge come segue:

Per informazioni, consulta Abilitazione delle implementazioni di terminazione durante gli eventi scale-in di Auto Scaling.

Come vengono CodeDeploy denominati i gruppi Amazon EC2 Auto Scaling

Durante blue/green le implementazioni su una piattaforma di EC2/On-Premises elaborazione, hai due opzioni per aggiungere istanze all'ambiente sostitutivo (verde):

- Utilizzare le istanze esistenti o create manualmente.
- Utilizza le impostazioni di un gruppo Amazon EC2 Auto Scaling che specifichi per definire e creare istanze in un nuovo gruppo Amazon EC2 Auto Scaling.

Se scegli la seconda opzione, effettua il CodeDeploy provisioning di un nuovo gruppo Amazon EC2 Auto Scaling per te. Viene utilizzata la seguente convenzione per assegnare il nome al gruppo:

```
CodeDeploy_deployment_group_name_deployment_id
```

Ad esempio, se una distribuzione con ID `10` implementa un gruppo di distribuzione denominato`alpha-deployments`, viene denominato il gruppo Amazon EC2 Auto Scaling fornito. `CodeDeploy_alpha-deployments_10` Per ulteriori informazioni, consultare Crea un gruppo di distribuzione per una distribuzione (console) EC2/On-Premises blue/green e GreenFleetProvisioningOption.

Ordine di esecuzione degli eventi hook del ciclo di vita personalizzati

Puoi aggiungere i tuoi hook del ciclo di vita ai gruppi di Amazon EC2 Auto Scaling a cui effettuare le distribuzioni. CodeDeploy Tuttavia, l'ordine in cui vengono eseguiti gli eventi del ciclo di vita personalizzati non può essere predeterminato in relazione agli eventi del ciclo di vita della distribuzione predefiniti. CodeDeploy Ad esempio, se aggiungi un hook del ciclo di vita personalizzato denominato `ReadyForSoftwareInstall` a un gruppo Amazon EC2 Auto Scaling, non puoi sapere in anticipo se verrà eseguito prima del primo o dopo l'ultimo evento del ciclo di vita di distribuzione predefinito. CodeDeploy

Per informazioni su come aggiungere hook personalizzati per il ciclo di vita a un gruppo Amazon EC2 Auto Scaling, consulta Adding lifecycle hook nella Amazon EC2 Auto Scaling User Guide.

Scale-out eventi durante una distribuzione

Se si verifica un evento di scalabilità orizzontale di Auto Scaling mentre è in corso una distribuzione, le nuove istanze verranno aggiornate con la revisione dell'applicazione precedentemente distribuita, non con la revisione dell'applicazione più recente. Se la distribuzione ha esito positivo, le vecchie istanze e le nuove istanze con scalabilità orizzontale ospiteranno diverse revisioni delle applicazioni. Per aggiornare le istanze con la revisione precedente, avvia CodeDeploy automaticamente una distribuzione successiva (subito dopo la prima) per aggiornare eventuali istanze obsolete. Se desideri modificare questo comportamento predefinito in modo che le istanze EC2 obsolete rimangano nella versione precedente, consulta. Automatic updates to outdated instances

Se desideri sospendere i processi di scalabilità orizzontale di Amazon EC2 Auto Scaling durante le distribuzioni, puoi farlo tramite un'impostazione nello script con cui viene utilizzato per il bilanciamento `common_functions.sh` del carico. CodeDeploy Se`HANDLE_PROCS=true`, i seguenti eventi di Auto Scaling vengono sospesi automaticamente durante il processo di distribuzione:

- AZRebalance
- AlarmNotification
- ScheduledActions
- ReplaceUnhealthy

> **Important**
>
> Solo la configurazione CodeDeployDefault.OneAtATime di distribuzione supporta questa funzionalità.

Per ulteriori informazioni sull'utilizzo per `HANDLE_PROCS=true` evitare problemi di distribuzione quando si utilizza Amazon EC2 Auto Scaling, consulta Avviso importante AutoScaling sulla gestione dei processi in aws-codedeploy-samples su. GitHub

Scale-in eventi durante una distribuzione

Se un gruppo Auto Scaling inizia a scalare mentre è in corso una CodeDeploy distribuzione su quel gruppo Auto Scaling, potrebbe verificarsi una situazione di competizione tra il processo di

terminazione (inclusi gli eventi del ciclo di vita della distribuzione di terminazione) e altri eventi del ciclo di vita sull'istanza di CodeDeploy terminazione. CodeDeploy La distribuzione su quell'istanza specifica potrebbe non riuscire se l'istanza viene terminata prima del completamento di tutti gli eventi del ciclo di vita. CodeDeploy Inoltre, l' CodeDeploy implementazione complessiva potrebbe fallire o meno, a seconda di come hai impostato l'impostazione Minimum healthy hosts nella configurazione di distribuzione.

Ordine degli eventi in AWS CloudFormation script cfn-init

Se si utilizza `cfn-init` (or`cloud-init`) per eseguire script su Linux-based istanze appena assegnate, le distribuzioni potrebbero fallire a meno che non si controlli rigorosamente l'ordine degli eventi che si verificano dopo l'avvio dell'istanza.

Questo ordine deve essere:

1. L'istanza appena assegnata viene avviata.
2. Tutti gli script di bootstrap `cfn-init` vengono eseguiti fino al completamento.
3. L'agente si avvia. CodeDeploy
4. La revisione più recente dell'applicazione viene distribuita all'istanza.

Se l'ordine degli eventi non è controllato con attenzione, l' CodeDeploy agente potrebbe avviare una distribuzione prima che tutti gli script siano terminati.

Per controllare l'ordine degli eventi, utilizzare una di queste best practice:

- Installa l' CodeDeploy agente tramite uno `cfn-init` script, inserendolo dopo tutti gli altri script.
- Includi l' CodeDeploy agente in un'AMI personalizzata e usa uno `cfn-init` script per avviarlo, inserendolo dopo tutti gli altri script.

Per informazioni sull'utilizzo*`cfn-init`*, consulta cfn-init nella Guida per l'utente.AWS CloudFormation

Utilizzo di un'AMI personalizzata con CodeDeploy Amazon EC2 Auto Scaling

Sono disponibili due opzioni per specificare l'AMI di base da utilizzare quando vengono lanciate nuove istanze Amazon EC2 in un gruppo Amazon EC2 Auto Scaling:

- È possibile specificare un'AMI personalizzata di base su cui sia già installato l' CodeDeploy agente. Poiché l'agente è già installato, questa opzione avvia nuove istanze Amazon EC2 più rapidamente

rispetto all'altra opzione. Tuttavia, questa opzione offre una maggiore probabilità che le distribuzioni iniziali delle istanze Amazon EC2 abbiano esito negativo, soprattutto se l' CodeDeploy agente non è aggiornato. Se scegli questa opzione, ti consigliamo di aggiornare regolarmente l' CodeDeploy agente nella tua AMI personalizzata di base.

- Puoi specificare un'AMI di base in cui non sia installato l' CodeDeploy agente e farlo installare ogni volta che una nuova istanza viene lanciata in un gruppo Amazon EC2 Auto Scaling. Sebbene questa opzione avvii nuove istanze Amazon EC2 più lentamente rispetto all'altra opzione, offre maggiori probabilità che le distribuzioni iniziali delle istanze abbiano successo. Questa opzione utilizza la versione più recente dell'agente CodeDeploy.

Integrazione CodeDeploy con Elastic Load Balancing

Durante CodeDeploy le implementazioni, un sistema di bilanciamento del carico impedisce che il traffico Internet venga indirizzato verso istanze che non sono pronte, sono attualmente in fase di implementazione o non sono più necessarie come parte di un ambiente. Il ruolo esatto svolto dal sistema di bilanciamento del carico, tuttavia, dipende dal fatto che venga utilizzato in una distribuzione o in loco. blue/green

> **Note**
>
> L'uso dei sistemi di bilanciamento del carico Elastic Load Balancing è obbligatorio nelle blue/green implementazioni e facoltativo nelle implementazioni sul posto.

Tipi di sistema di bilanciamento del carico elastico

Elastic Load Balancing offre tre tipi di sistemi di bilanciamento del carico che possono essere utilizzati nelle CodeDeploy distribuzioni: Classic Load Balancer, Application Load Balancer e Network Load Balancer.

Classic Load Balancer

Percorsi e bilanci di carico a livello di trasporto () o a livello di applicazione (). TCP/SSL HTTP/HTTPS Supporta un VPC.

> **Note**
>
> I sistemi Classic Load Balancer non sono supportati nelle distribuzioni di Amazon ECS.

Application Load Balancer

Instrada e bilancia il carico a livello di applicazione () HTTP/HTTPS e supporta il routing basato sui percorsi. È possibile instradare le richieste alle porte su ogni istanza EC2 o istanza di container nel virtual private cloud.

> **Note**
>
> I gruppi target di Application Load Balancer devono avere un tipo di destinazione per le implementazioni su istanze EC2 e `instance` per le implementazioni Fargate. `IP` Per ulteriori informazioni, consulta Target type.

Network Load Balancer

I percorsi e i bilanci di carico a livello di trasporto (TCP/UDP Layer-4) si basano sulle informazioni sugli indirizzi estratte dall'intestazione del pacchetto TCP, non dal contenuto del pacchetto. Il Network Load Balancer è in grado di gestire i picchi di traffico, mantenere l'IP sorgente del client e utilizzare un IP fisso per l'intera durata del sistema di bilanciamento del carico.

Per ulteriori informazioni sui sistemi di bilanciamento del carico Elastic Load Balancing, consulta i seguenti argomenti:

- Cos'è l'Elastic Load Balancing?
- Cos'è un Classic Load Balancer?
- Cos'è un Application Load Balancer?
- Cos'è un Network Load Balancer?

Blue/Green distribuzioni

Il reindirizzamento del traffico delle istanze dietro un sistema di bilanciamento del carico Elastic Load Balancing è fondamentale per le implementazioni. CodeDeploy blue/green

Durante una blue/green distribuzione, il load balancer consente di indirizzare il traffico verso le nuove istanze in un gruppo di distribuzione in cui è stata distribuita la revisione più recente dell'applicazione (l'ambiente sostitutivo), in base alle regole specificate, e quindi blocca il traffico proveniente dalle vecchie istanze in cui era in esecuzione la revisione precedente dell'applicazione (l'ambiente originale).

Dopo la registrazione delle istanze in un ambiente sostitutivo con uno o più sistemi di bilanciamento del carico, le istanze dell'ambiente originale vengono annullate e, se lo si desidera, terminate.

Per una blue/green distribuzione, puoi specificare uno o più gruppi target Classic Load Balancer, Application Load Balancer o Network Load Balancer nel tuo gruppo di distribuzione. È possibile utilizzare la CodeDeploy console o aggiungere AWS CLI i load balancer a un gruppo di distribuzione.

Per ulteriori informazioni sui sistemi di bilanciamento del carico nelle blue/green distribuzioni, consulta i seguenti argomenti:

- Configura un sistema di bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2
- Crea un'applicazione per una blue/green distribuzione (console)
- Crea un gruppo di distribuzione per una distribuzione (console) EC2/On-Premises blue/green

In-place distribuzioni

Durante una distribuzione in loco, un sistema di bilanciamento del carico impedisce che il traffico Internet venga instradato a un'istanza che è in fase di distribuzione, quindi rende l'istanza nuovamente disponibile per il traffico una volta completata la distribuzione a quell'istanza.

Se un sistema di bilanciamento del carico non viene utilizzato durante una distribuzione in loco, il traffico Internet potrebbe comunque essere indirizzato a un'istanza durante il processo di distribuzione. Di conseguenza, le applicazioni Web potrebbero risultare ai clienti non funzionanti, incomplete o non aggiornate. Quando si utilizza un sistema di bilanciamento del carico Elastic Load Balancing con una distribuzione sul posto, le istanze di un gruppo di distribuzione vengono cancellate dal sistema di bilanciamento del carico, aggiornate con la revisione più recente dell'applicazione e quindi registrate nuovamente presso il sistema di bilanciamento del carico come parte dello stesso gruppo di distribuzione una volta completata la distribuzione. CodeDeploy aspetterà fino a 1 ora prima che l'istanza diventi integra dopo il bilanciamento del carico. Se l'istanza non viene contrassegnata come integra dal load balancer durante il periodo di attesa, passa CodeDeploy all'istanza successiva o fallisce la distribuzione, in base alla configurazione di distribuzione.

Per una distribuzione sul posto, puoi specificare uno o più gruppi target Classic Load Balancer, Application Load Balancer o Network Load Balancer. È possibile specificare i load balancer come parte della configurazione del gruppo di distribuzione oppure utilizzare uno script fornito da per implementare i load balancer. CodeDeploy

Specificate il sistema di bilanciamento del carico di distribuzione sul posto utilizzando un gruppo di distribuzione

Per aggiungere sistemi di bilanciamento del carico a un gruppo di distribuzione, si utilizza la console o. CodeDeploy AWS CLI Per informazioni su come specificare un sistema di bilanciamento del carico in un gruppo di distribuzione per distribuzioni in loco, consulta i seguenti argomenti:

- Crea un'applicazione per un'implementazione sul posto (console)
- Crea un gruppo di distribuzione per una distribuzione sul posto (console)
- Configura un sistema di bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2

Specificare il sistema di bilanciamento del carico di distribuzione sul posto utilizzando uno script

Utilizza la procedura descritta di seguito per usare script per il ciclo di vita delle distribuzioni per configurare un sistema di bilanciamento del carico per distribuzioni in loco.

> **Note**
>
> È necessario utilizzare la configurazione di CodeDeployDefault.OneAtATime distribuzione solo quando si utilizza uno script per configurare un sistema di bilanciamento del carico per una distribuzione sul posto. Le esecuzioni simultanee non sono supportate e l' CodeDeployDefault.OneAtATime impostazione garantisce l'esecuzione seriale degli script. Per ulteriori dettagli sulle configurazioni della distribuzione, consulta Utilizzo delle configurazioni di distribuzione in CodeDeploy.

Nel repository CodeDeploy Samples su GitHub, forniamo istruzioni ed esempi che puoi adattare per utilizzare i sistemi di bilanciamento del carico CodeDeploy Elastic Load Balancing. Questi repository includono tre script di esempio: `register_with_elb.sh`deregister_from_elb.sh, e `common_functions.sh` —che forniscono tutto il codice necessario per iniziare. È sufficiente modificare i segnaposto in questi tre script e fare riferimento agli script dal file `appspec.yml`.

Per configurare distribuzioni sul posto con istanze Amazon CodeDeploy EC2 registrate con sistemi di bilanciamento del carico Elastic Load Balancing, procedi come segue:

1. Scarica gli esempi per il tipo di sistema di bilanciamento del carico che desideri utilizzare per una distribuzione in loco:

- Classic Load Balancer
- Application Load Balancer o Network Load Balancer (lo stesso script può essere usato per entrambi i tipi)

2. Assicurati che su ciascuna delle tue istanze Amazon EC2 di destinazione sia installata. AWS CLI
3. Assicurati che ciascuna delle tue istanze Amazon EC2 di destinazione abbia un profilo di istanza IAM associato almeno alle autorizzazioni elasticloadbalancing: * e autoscaling: *.
4. Includi nella directory del codice sorgente dell'applicazione gli script degli eventi del ciclo di vita della distribuzione (`register_with_elb.sh`, `deregister_from_elb.sh` e `common_functions.sh`).
5. Nella revisione `appspec.yml` per l'applicazione, fornisci le istruzioni per eseguire lo script durante l'evento e lo script durante CodeDeploy l'evento. `register_with_elb.sh` ApplicationStart**`deregister_from_elb.sh`**ApplicationStop
6. Se l'istanza fa parte di un gruppo Amazon EC2 Auto Scaling, puoi saltare questo passaggio.

   Nello script `common_functions.sh`:

   - Se utilizzi Classic Load Balancer, specifica i nomi dei sistemi di bilanciamento del carico Elastic Load Balancing `ELB_LIST=""` in e apporta le modifiche necessarie alle altre impostazioni di distribuzione nel file.
   - Se utilizzi Application Load Balancer o Network Load Balancer, specifica i nomi dei gruppi target di Elastic Load Balancing `TARGET_GROUP_LIST=""` in e apporta le modifiche necessarie alle altre impostazioni di distribuzione nel file.
7. Raggruppa il codice sorgente dell'applicazione, `appspec.yml` e gli script degli eventi del ciclo di vita della distribuzione in una revisione dell'applicazione, quindi carica quest'ultima. Implementa la revisione nelle istanze Amazon EC2. Durante la distribuzione, gli script degli eventi del ciclo di vita della distribuzione annulleranno la registrazione dell'istanza Amazon EC2 con il sistema di bilanciamento del carico, attenderanno che la connessione si esaurisca e quindi registreranno nuovamente l'istanza Amazon EC2 con il sistema di bilanciamento del carico al termine della distribuzione.

Integrazione con prodotti e servizi dei partner

CodeDeploy dispone di un'integrazione integrata per i seguenti prodotti e servizi dei partner:

Ansible	Se disponi già di un set di playbook Ansible, ma hai solo bisogno di un posto dove eseguirli , il modello per Ansible CodeDeploy dimostra come un paio di semplici hook di implement azione possano garantire che Ansible sia disponibile sull'istanza di distribuzione locale ed esegua i playbook. Se disponi già di un processo per la creazione e la manutenzione del tuo inventario, c'è anche un modulo Ansible che puoi usare per installare ed eseguire l'agente. CodeDeploy Ulteriori informazioni: • Ansible e CodeDeploy
Atlassian — Bamboo e Bitbucket	Il CodeDeploy task per Bamboo comprime la directory che contiene un AppSpec file in un file.zip, carica il file su Amazon S3 e quindi avvia la distribuzione in base alla configura zione fornita nell'applicazione. CodeDeploy Il supporto di Atlassian Bitbucket ti CodeDeplo y consente di inviare codice alle istanze Amazon EC2 direttamente dall'interfaccia utente Bitbucket, su richiesta, a qualsiasi gruppo di distribuzione. Ciò significa che dopo aver aggiornato il codice nel tuo repository Bitbucket, non devi accedere alla tua piattafor ma di integrazione continua (CI) o alle istanze Amazon EC2 per eseguire un processo di distribuzione manuale. Ulteriori informazioni: • Utilizzo del task per Bamboo CodeDeploy

	• Annuncio del supporto di Atlassian Bitbucket per CodeDeploy
Chef	AWS fornisce due esempi di modelli per l'integrazione di Chef e. CodeDeploy Il primo è un ricettario di Chef che installa e avvia l'agente. CodeDeploy Ciò ti consente di continuare a gestire la tua infrastruttura host con Chef durante l'utilizzo di CodeDeploy. Il secondo modello di esempio mostra come CodeDeploy orchestrare l'esecuzione di libri di cucina e ricette con chef-solo su ciascun nodo. Ulteriori informazioni: • Chef e CodeDeploy
CircleCI	CircleCI fornisce un set di strumenti per i test automatizzati e per l'integrazione e la distribuz ione continua. Dopo aver creato un ruolo IAM AWS da utilizzare con CircleCI e aver configura to i parametri di distribuzione nel file circle.ym l, puoi usare CircleCI con per creare revisioni delle applicazioni, caricarle in un CodeDeploy bucket Amazon S3 e quindi avviare e monitorar e le distribuzioni. Ulteriori informazioni: • Utilizzo di un CircleCI Orb per distribuire applicazioni su AWS CodeDeploy

CloudBees	È possibile utilizzare il plug-in CodeDeploy Jenkins, disponibile su CloudBees DEV @cloud, come azione post-compilazione. Ad esempio, al termine di una pipeline di distribuzione continua, puoi utilizzarlo per distribuire la revisione di un'applicazione al tuo parco istanze dei server. Ulteriori informazioni: • CodeDeploy Il plugin Jenkins è ora disponibile su DEV @cloud
Codeship	È possibile utilizzare Codeship per distribuire le revisioni delle applicazioni tramite. CodeDeploy Puoi utilizzare l'interfaccia utente di Codeship per aggiungere CodeDeploy a una pipeline di distribuzione per un ramo. Ulteriori informazioni: • Esegui la distribuzione su CodeDeploy • CodeDeploy integrazione su Codeship
GitHub	È possibile utilizzare CodeDeploy per distribuire le revisioni delle applicazioni dai repository. GitHub È inoltre possibile attivare una distribuzione da un GitHub repository ogni volta che il codice sorgente in tale repository viene modificato. Ulteriori informazioni: • Integrazione con CodeDeploy GitHub • Tutorial: utilizzare CodeDeploy per distribuire un'applicazione da GitHub • Implementa automaticamente dall'utilizzo GitHub CodeDeploy

HashiCorp Console	È possibile utilizzare lo strumento open source HashiCorp Consul per contribuire a garantire l'integrità e la stabilità dell'ambiente applicativo quando si distribuiscono le applicazioni in. CodeDeploy Puoi utilizzare Consul per registrare le applicazioni in modo che vengano rilevate durante la distribuzione, inserire le applicazioni e i nodi in modalità di manutenzione per escluderli dalle distribuzioni e interrompere le distribuzioni nel caso in cui le istanze di destinazione presentino problemi di integrità. Ulteriori informazioni: • CodeDeploy distribuzioni con Consul HashiCorp
Jenkins	Il plugin CodeDeploy Jenkins fornisce una fase di post-compilazione per il tuo progetto Jenkins. Una volta completata con successo, comprime l'area di lavoro, carica su Amazon S3 e avvia una nuova distribuzione. Ulteriori informazioni: • CodeDeployPlugin Jenkins • Configurazione del plugin Jenkins per CodeDeploy

Puppet Labs	AWS fornisce modelli di esempio per Puppet e. CodeDeploy Il primo è un modulo Puppet che installa e avvia l'agente. CodeDeploy Ciò ti consente di continuare a gestire la tua infrastruttura host con Puppet durante l'utilizzo di CodeDeploy. Il secondo modello di esempio mostra come CodeDeploy orchestrare l'esecuzi one di moduli e manifesti con un pupazzo masterless su ogni nodo. Ulteriori informazioni: • Puppet e CodeDeploy
SaltStack	Puoi integrare l'SaltStackinfrastruttura con CodeDeploy. Potete usare il CodeDeploy modulo per installare ed eseguire l' CodeDeplo y agente sui vostri servitori oppure, con un paio di semplici hook di distribuzione, potete usarlo CodeDeploy per orchestrare il funzionamento dei vostri Salt States. Ulteriori informazioni: • SaltStack e CodeDeploy

TeamCity	È possibile utilizzare il plug-in CodeDeploy Runner per distribuire applicazioni direttamente da. TeamCity Il plug-in aggiunge una fase di TeamCity compilazione che prepara e carica una revisione dell'applicazione in un bucket Amazon S3, registra la revisione in un' CodeDeployapplicazione, crea una CodeDeploy distribuzione e, se lo desideri, attende il completamento della distribuzione. Ulteriori informazioni: • CodeDeploy Runner (Scarica) • CodeDeploy Plugin Runner (documentazione)
Travis CI	È possibile configurare Travis CI per attivare una distribuzione CodeDeploy dopo una compilazione riuscita. Ulteriori informazioni: • Travis CI e implementazioni CodeDeploy

Argomenti

- Integrazione con CodeDeploy GitHub

Integrazione con CodeDeploy GitHub

CodeDeploy supporta GitHub, un servizio di hosting e condivisione di codice basato sul Web. CodeDeploy può distribuire le revisioni delle applicazioni archiviate in GitHub repository o bucket Amazon S3 sulle istanze. CodeDeploy GitHub supporta solo le EC2/On-Premises distribuzioni.

Argomenti

- Distribuzione delle revisioni da CodeDeploy GitHub
- GitHub comportamenti con CodeDeploy

Distribuzione delle revisioni da CodeDeploy GitHub

Per distribuire una revisione dell'applicazione da un GitHub repository alle istanze:

1. Crea una revisione compatibile con CodeDeploy il tipo di istanza Amazon EC2 su cui effettuerai la distribuzione.

   Per creare una revisione compatibile, seguire le istruzioni in Pianifica una revisione per CodeDeploy e Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy.
2. Usa un GitHub account per aggiungere la revisione a un repository. GitHub

   Per creare un GitHub account, vedi Join. GitHub Per creare un GitHub repository, consulta Creare un repository.
3. Utilizza la pagina Crea distribuzione nella CodeDeploy console o il AWS CLI create-deployment comando per distribuire la revisione dal GitHub repository alle istanze di destinazione configurate per l'uso nelle distribuzioni. CodeDeploy

   Se desideri chiamare il create-deployment comando, devi prima utilizzare la pagina Crea distribuzione della console per CodeDeploy autorizzare l'interazione con GitHub l'account preferito GitHub per l'applicazione specificata. Dovrai eseguire questa operazione una sola volta per ogni applicazione.

   Per informazioni su come utilizzare la pagina Crea distribuzione per la distribuzione da un GitHub repository, consulta. Crea una distribuzione con CodeDeploy

   Per informazioni su come chiamare il create-deployment comando per la distribuzione da un GitHub repository, consulta. Crea un'implementazione di EC2/On-Premises Compute Platform (CLI)

   Per informazioni su come preparare le istanze da utilizzare nelle CodeDeploy distribuzioni, consulta. Utilizzo delle istanze per CodeDeploy

Per ulteriori informazioni, consulta Tutorial: utilizzare CodeDeploy per distribuire un'applicazione da GitHub.

GitHub comportamenti con CodeDeploy

Argomenti

- GitHub autenticazione con applicazioni in CodeDeploy

- CodeDeploy interazione con archivi privati e pubblici GitHub
- CodeDeploy interazione con gli archivi gestiti dall'organizzazione GitHub
- Implementa automaticamente da CodePipeline con CodeDeploy

GitHub autenticazione con applicazioni in CodeDeploy

Dopo aver concesso CodeDeploy l'autorizzazione all'interazione GitHub, l'associazione tra l' GitHub account e l'applicazione viene archiviata in CodeDeploy. È possibile collegare l'applicazione a un altro GitHub account. Puoi anche revocare l'autorizzazione CodeDeploy all'interazione con. GitHub

Per collegare un GitHub account a un'applicazione in CodeDeploy

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Scegli l'applicazione che desideri collegare a un altro GitHub account.
4. Se l'applicazione non dispone di un gruppo di distribuzione, scegli Crea gruppo di distribuzione per crearne uno. Per ulteriori informazioni, consulta Crea un gruppo di distribuzione con CodeDeploy. Un gruppo di distribuzione deve scegliere Create deployment (Crea distribuzione) nella fase successiva.
5. Da Deployments (Distribuzioni), scegli Create deployment (Crea distribuzione).

   > **Note**
   >
   > Non è necessario creare una nuova distribuzione. Questo è attualmente l'unico modo per collegare un GitHub account diverso a un'applicazione.

6. Nelle impostazioni di distribuzione, per Tipo di revisione, scegli La mia applicazione è archiviata in GitHub.
7. Esegui una delle seguenti operazioni:

- Per creare una connessione per AWS CodeDeploy le applicazioni a un GitHub account, esci da una scheda separata del browser web. GitHub In Nome GitHub token, digita un nome per identificare la connessione, quindi scegli Connetti a GitHub. La pagina Web richiede l'autorizzazione CodeDeploy all'interazione con GitHub l'applicazione. Continuare con la fase 10.
- Per utilizzare una connessione già creata, in Nome GitHub token, selezionane il nome, quindi scegli Connetti a GitHub. Continuare con la fase 8.
- Per creare una connessione a un altro GitHub account, esci da una scheda separata del browser web. GitHub In Nome GitHub token, digita un nome per identificare la connessione, quindi scegli Connetti a GitHub. La pagina Web richiede l'autorizzazione CodeDeploy all'interazione con GitHub l'applicazione. Continuare con la fase 10.

8. Se non hai già effettuato l'accesso GitHub, segui le istruzioni nella pagina di accesso per accedere con l' GitHub account a cui desideri collegare l'applicazione.
9. Scegli Autorizza applicazione. GitHub dà CodeDeploy il permesso di interagire con GitHub l' GitHub account con cui è stato effettuato l'accesso all'applicazione selezionata.
10. Se non si desidera creare una distribuzione, scegliere Cancel (Annulla).

Per revocare l'autorizzazione all'interazione con CodeDeploy GitHub

1. Accedi GitHub utilizzando le credenziali per l' GitHub account in cui desideri revocare l'autorizzazione. AWS CodeDeploy
2. Apri la pagina GitHub Applicazioni, CodeDeployindividua nell'elenco delle applicazioni autorizzate, quindi segui la GitHub procedura per revocare l'autorizzazione per un'applicazione.

CodeDeploy interazione con archivi privati e pubblici GitHub

CodeDeploy supporta la distribuzione di applicazioni da GitHub archivi privati e pubblici. Quando CodeDeploy autorizzi l'accesso per tuo GitHub conto, CodeDeploy avrai accesso in lettura e scrittura a tutti gli GitHub archivi privati a cui ha accesso il tuo GitHub account. Tuttavia, legge CodeDeploy solo dai repository. GitHub Non scriverà su nessuno dei tuoi repository privati GitHub .

CodeDeploy interazione con gli archivi gestiti dall'organizzazione GitHub

Per impostazione predefinita, gli GitHub archivi gestiti da un'organizzazione (a differenza degli archivi privati o pubblici dell'account) non concedono l'accesso ad applicazioni di terze parti, tra cui. CodeDeploy La distribuzione avrà esito negativo se le restrizioni relative alle applicazioni di terze

parti di un'organizzazione sono abilitate GitHub e si tenta di distribuire codice dal relativo repository. GitHub Ci sono due modi per risolvere questo problema.

- In qualità di membro di un'organizzazione, puoi richiedere al proprietario dell'organizzazione di approvare l'accesso ad CodeDeploy. I passaggi per richiedere questo accesso dipendono dal fatto che tu abbia già autorizzato il tuo account CodeDeploy individuale:
  - Se disponi dell'accesso autorizzato a CodeDeploy nel tuo account, vedi Richiesta dell'approvazione dell'organizzazione per le applicazioni autorizzate.
  - Se non hai ancora autorizzato l'accesso al CodeDeploy tuo account, vedi Richiesta dell'approvazione dell'organizzazione per le applicazioni di terze parti.
- Il proprietario dell'organizzazione può disabilitare tutte le restrizioni delle applicazioni di terze parti per l'organizzazione. Per informazioni, consulta Disattivazione delle restrizioni relative alle applicazioni di terze parti per l'organizzazione.

Per ulteriori informazioni, consulta Informazioni sulle restrizioni delle applicazioni di terze parti.

Implementa automaticamente da CodePipeline con CodeDeploy

È possibile attivare una distribuzione da un CodePipeline ogni volta che il codice sorgente cambia. Per ulteriori informazioni, consulta CodePipeline.

Esempi di integrazione dalla community

Le seguenti sezioni forniscono collegamenti a post di blog, articoli ed esempi della community.

> **Note**
>
> Questi link vengono forniti esclusivamente a scopo informativo e non devono essere considerati un elenco completo o un'approvazione dei contenuti degli esempi. AWS declina qualsiasi responsabilità relativamente ai contenuti o alla precisione dei contenuti esterni.

Post del blog

- Automatizzazione del provisioning in CodeDeploy CloudFormation

  Scopri come effettuare il provisioning della distribuzione di un'applicazione CodeDeploy utilizzando. CloudFormation

Pubblicazione: gennaio 2016

- AWS Toolkit for Eclipse Integrazione con CodeDeploy (Parte 1)

  AWS Toolkit for Eclipse Integrazione con CodeDeploy (Parte 2)

  AWS Toolkit for Eclipse Integrazione con CodeDeploy (Parte 3)

  Scopri come gli sviluppatori Java possono utilizzare il CodeDeploy plug-in per Eclipse per distribuire applicazioni Web AWS direttamente dagli ambienti di sviluppo Eclipse.

  Pubblicazione: febbraio 2015

- Implementa automaticamente utilizzando GitHub CodeDeploy

  Scopri come utilizzare le implementazioni automatiche da GitHub a per CodeDeploy creare una pipeline completa, dal controllo del codice sorgente agli ambienti di test o produzione.

  Pubblicazione: dicembre 2014

CodeDeploy tutorial

Questa sezione include alcuni tutorial per imparare a utilizzare CodeDeploy.

Le procedure di questi tutorial forniscono suggerimenti sulla posizione in cui archiviare i file (ad esempio, c:\temp) e i nomi da assegnare a bucket, sottocartelle o file (ad esempio, amzn-s3-demo-bucket e, rispettivamente), ma non è necessario utilizzarle. HelloWorldApp CodeDeployDemo-EC2-Trust.json Assicurati di sostituire nomi e percorsi dei file durante l'esecuzione delle procedure.

Argomenti

- Tutorial: Esegui WordPress la distribuzione su un'istanza Amazon EC2 (Amazon Linux o Red Hat Enterprise Linux e Linux, macOS o Unix)
- Tutorial: distribuisci un messaggio «ciao, mondo!» applicazione con CodeDeploy (Windows Server)
- Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux)
- Tutorial: CodeDeploy Da utilizzare per distribuire un'applicazione in un gruppo di Auto Scaling
- Tutorial: utilizzare CodeDeploy per distribuire un'applicazione da GitHub
- Tutorial: distribuire un'applicazione in Amazon ECS
- Tutorial: distribuisci un servizio Amazon ECS con un test di convalida
- Tutorial: Implementa una funzione CodeDeploy Lambda aggiornata con e AWS Modello di applicazione serverless

Tutorial: Esegui WordPress la distribuzione su un'istanza Amazon EC2 (Amazon Linux o Red Hat Enterprise Linux e Linux, macOS o Unix)

In questo tutorial, WordPress distribuisci uno strumento di blogging open source e un sistema di gestione dei contenuti basato su PHP e MySQL su una singola istanza Amazon EC2 che esegue Amazon Linux o Red Hat Enterprise Linux (RHEL).

Non hai trovato ciò che cerchi?

- Per fare pratica con la distribuzione su un'istanza Amazon EC2 che esegue invece Windows Server, consulta. Tutorial: distribuisci un messaggio «ciao, mondo!» applicazione con CodeDeploy (Windows Server)
- Per fare pratica con la distribuzione su un'istanza locale anziché su un'istanza Amazon EC2, consulta. Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux)

I passaggi di questo tutorial sono presentati dal punto di vista di una macchina di sviluppo locale che esegue Linux, macOS o Unix. Sebbene sia possibile completare buona parte della procedura su un computer locale che esegue Windows, occorrerà adattare le fasi che includono comandi quali chmod e wget, applicazioni come sed e percorsi di directory quali `/tmp`.

Prima di avviare questo tutorial, è necessario completare i prerequisiti in Guida introduttiva con CodeDeploy. Questi includono la configurazione di un utente, l'installazione o l'aggiornamento e la AWS CLI creazione di un profilo di istanza IAM e di un ruolo di servizio.

Argomenti

- Fase 1: Avvio e configurazione di un'istanza Amazon Linux o Red Hat Enterprise Linux Amazon EC2
- Fase 2: configura il contenuto sorgente da distribuire sull'istanza Amazon Linux o Red Hat Enterprise Linux Amazon EC2
- Passaggio 3: carica l' WordPress applicazione su Amazon S3
- Fase 4: Implementazione dell'applicazione WordPress
- Fase 5: Aggiornare e ridistribuire l'applicazione WordPress
- Passaggio 6: Pulisci WordPress l'applicazione e le risorse correlate

Fase 1: Avvio e configurazione di un'istanza Amazon Linux o Red Hat Enterprise Linux Amazon EC2

Per distribuire l' WordPress applicazione CodeDeploy, avrai bisogno di un'istanza Amazon EC2 che esegua Amazon Linux o Red Hat Enterprise Linux (RHEL). L'istanza Amazon EC2 richiede una nuova regola di sicurezza in entrata che consenta le connessioni HTTP. Questa regola è necessaria per visualizzare la WordPress pagina in un browser dopo che è stata distribuita correttamente.

Segui le istruzioni in Crea un'istanza Amazon EC2 per CodeDeploy. Quando arrivi alla parte di queste istruzioni sull'assegnazione di un tag di istanza Amazon EC2 all'istanza, assicurati di specificare la chiave del tag e il valore **`Name`** del tag di. **`CodeDeployDemo`** (Se viene specificata una chiave o un valore del tag, le istruzioni in Fase 4: Implementazione dell'applicazione WordPress possono portare a risultati inaspettati).

Dopo aver seguito le istruzioni per avviare l'istanza Amazon EC2, torna a questa pagina e passa alla sezione successiva. Non continuare con Crea un'applicazione con CodeDeploy il passaggio successivo.

Connect alla tua istanza Amazon Linux o RHEL Amazon EC2

Dopo il lancio della tua nuova istanza Amazon EC2, segui queste istruzioni per esercitarti a connetterti ad essa.

1. Usa il ssh comando (o un emulatore di SSH-capable terminale come PuTTY) per connetterti alla tua istanza Amazon Linux o RHEL Amazon EC2. Avrai bisogno dell'indirizzo DNS pubblico dell'istanza e della chiave privata per la coppia di chiavi utilizzata all'avvio dell'istanza Amazon EC2. Per ulteriori informazioni, consulta Connect to Your Instance.

   Ad esempio, se l'indirizzo DNS pubblico è **`ec2-01-234-567-890.compute-1.amazonaws.com`** e la coppia di chiavi dell'istanza Amazon EC2 per l'accesso SSH è **`codedeploydemo.pem`** denominata, devi digitare:

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

   Sostituisci */path/to/*`codedeploydemo.pem` con il percorso del `.pem` file e l'indirizzo DNS di esempio con l'indirizzo della tua istanza Amazon Linux o RHEL Amazon EC2.

   > **Note**
   >
   > Se si riceve un errore riguardo al fatto che le autorizzazioni della chiave del file sono troppo aperte, sarà necessario limitare le autorizzazioni per dare accesso solo all'utente attuale (tu). Ad esempio, con il chmod comando su Linux, macOS o Unix, digitare:

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

2. Dopo aver effettuato l'accesso, vedrai il banner AMI per l'istanza Amazon EC2. Per Amazon Linux, dovrebbe avere il seguente aspetto:

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

3. Ora puoi disconnetterti dall'istanza Amazon EC2 in esecuzione.

> **Warning**
>
> Non interrompere o terminare l'istanza Amazon EC2. In caso contrario, CodeDeploy non sarà possibile eseguire la distribuzione su di essa.

Aggiungi una regola in entrata che consenta il traffico HTTP verso la tua istanza Amazon Linux o RHEL Amazon EC2

Il passaggio successivo conferma che l'istanza Amazon EC2 ha una porta HTTP aperta in modo da poter visualizzare la home page dell' WordPress applicazione distribuita in un browser.

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Scegli Istanze, quindi scegli la tua istanza.
3. Nella scheda Descrizione, in Gruppi di sicurezza, scegli Visualizza regole in entrata.

   Dovresti vedere un elenco di regole nel tuo gruppo di sicurezza come il seguente:

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

4. In Gruppi di sicurezza, scegli il gruppo di sicurezza per la tua istanza Amazon EC2. Potrebbe essere denominato **`launch-wizard-`*`N`***. Il nome ***`N`*** è un numero assegnato al proprio gruppo di sicurezza quando è stata creata l'istanza.

   Seleziona la scheda Inbound (In entrata). Se il gruppo di sicurezza per la tua istanza è configurato correttamente, dovresti vedere una regola con i seguenti valori:

- Type (Tipo): HTTP
- Protocollo: TCP
- Port Range (Intervallo porte): 80
- Fonte: 0.0.0. 0/0

5. Se non vedi una regola con questi valori, usa le procedure in Aggiungere regole a un gruppo di sicurezza per aggiungerle a una nuova regola di sicurezza.

Fase 2: configura il contenuto sorgente da distribuire sull'istanza Amazon Linux o Red Hat Enterprise Linux Amazon EC2

È ora possibile configurare l'applicazione dei contenuti di origine in modo da avere materiale da distribuire all'istanza.

Argomenti

- Ottieni il codice sorgente
- Crea script per eseguire la tua applicazione
- Aggiungere un file di specifiche dell'applicazione

Ottieni il codice sorgente

In questo tutorial, distribuisci la piattaforma di pubblicazione WordPress dei contenuti dalla tua macchina di sviluppo all'istanza Amazon EC2 di destinazione. Per ottenere il codice WordPress sorgente, puoi utilizzare chiamate da riga di comando integrate. In alternativa, si può usare Git se è installato nel computer di sviluppo.

Per questi passaggi, supponiamo che tu abbia scaricato una copia del codice WordPress sorgente `/tmp` nella directory del tuo computer di sviluppo. È possibile scegliere una directory qualsiasi, ma ci si deve ricordare di sostituire la posizione per `/tmp` nel punto specificato in queste fasi.

Scegli una delle due opzioni seguenti per copiare i file WordPress sorgente sulla tua macchina di sviluppo. La prima opzione utilizza le chiamate della riga di comando integrata. La seconda usa Git.

Argomenti

- Per ottenere una copia del codice WordPress sorgente (chiamate da riga di comando integrate)
- Per ottenere una copia del codice WordPress sorgente (Git)

Per ottenere una copia del codice WordPress sorgente (chiamate da riga di comando integrate)

1. Chiamate il wget comando per scaricare una copia del codice WordPress sorgente, come file.zip, nella directory corrente:

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

2. Chiamare i comandi unzip, mkdir, cp e rm per:

   - Decomprimi il `master.zip` file nella `/tmp/WordPress_Temp` directory (cartella).
   - Copiare i contenuti estratti dallo zip nella cartella di destinazione `/tmp/WordPress`.
   - Eliminare la cartella temporanea `/tmp/WordPress_Temp` e il file `master`.

   Eseguire uno alla volta i comandi:

   ```
   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
   ```

   Questo ti lascia con un set pulito di file di codice WordPress sorgente nella `/tmp/WordPress` cartella.

Per ottenere una copia del codice WordPress sorgente (Git)

1. Scaricare e installare Git sul computer di sviluppo.
2. Nella cartella `/tmp/WordPress`, chiamare il comando git init.
3. Chiamate il git clone comando per clonare il WordPress repository pubblico, creandone una copia nella cartella di `/tmp/WordPress` destinazione:

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

Questo ti lascia con un set pulito di file di codice WordPress sorgente nella `/tmp/WordPress` cartella.

Crea script per eseguire la tua applicazione

Quindi, crea una cartella e degli script nella directory. CodeDeploy utilizza questi script per configurare e distribuire la revisione dell'applicazione sull'istanza Amazon EC2 di destinazione. È possibile usare qualsiasi editor di testo per creare gli script.

1. Crea una directory di script nella tua copia del codice sorgente: WordPress

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

2. Creare un file `install_dependencies.sh` in `/tmp/WordPress/scripts`. Aggiungere le seguenti righe al file. Questo script `install_dependencies.sh` installa Apache, MySQL e PHP. Aggiunge inoltre il supporto MySQL a PHP.

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

3. Creare un file `start_server.sh` in `/tmp/WordPress/scripts`. Aggiungere le seguenti righe al file. Questo script `start_server.sh` avvia Apache e MySQL.

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

4. Creare un file `stop_server.sh` in `/tmp/WordPress/scripts`. Aggiungere le seguenti righe al file. Questo script `stop_server.sh` arresta Apache e MySQL.

   ```
   #!/bin/bash
   isExistApp="pgrep httpd"
   if [[ -n $isExistApp ]]; then
   ```

```
systemctl stop httpd.service
fi
isExistApp=pgrep mysqld
if [[ -n $isExistApp ]]; then
systemctl stop mariadb.service
fi
isExistApp=pgrep php-fpm
if [[ -n $isExistApp ]]; then
systemctl stop php-fpm.service

fi
```

5. Creare un file `create_test_db.sh` in `/tmp/WordPress/scripts`. Aggiungere le seguenti righe al file. Questo `create_test_db.sh` script utilizza MySQL per creare **`test`** un database WordPress da utilizzare.

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

6. Infine creare uno script `change_permissions.sh` in `/tmp/WordPress/scripts`. Questo viene utilizzato per modificare le autorizzazioni per la cartella in Apache.

> **Important**
>
> Questo script ha aggiornato le autorizzazioni nella cartella `/tmp/WordPress` in modo che tutti possano scriverci. Questo è necessario per WordPress poter scrivere nel suo database durante. Fase 5: Aggiornare e ridistribuire l'applicazione WordPress Dopo aver configurato l' WordPress applicazione, esegui il comando seguente per aggiornare le autorizzazioni a un'impostazione più sicura:
>
> ```
> chmod -R 755 /var/www/html/WordPress
> ```

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

7. Assegnare autorizzazioni eseguibili a tutti gli script. Sulla riga di comando, digitare:

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

Aggiungere un file di specifiche dell'applicazione

Quindi, aggiungete un file di specifiche dell'applicazione (AppSpec file), un file in formato YAML utilizzato da per: CodeDeploy

- Mappa i file di origine nella revisione dell'applicazione alle rispettive destinazioni sull'istanza Amazon EC2 di destinazione.
- Specificare autorizzazioni personalizzate per i file distribuiti.
- Specificare gli script da eseguire sull'istanza Amazon EC2 di destinazione durante la distribuzione.

Il AppSpec file deve avere un nome. `appspec.yml` Deve essere inserito nella directory root del codice sorgente dell'applicazione. In questo tutorial, la directory root è `/tmp/WordPress`.

Con un editor di testo, creare un file denominato `appspec.yml`. Aggiungere le seguenti righe al file:

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/change_permissions.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
    - location: scripts/create_test_db.sh
      timeout: 300
      runas: root
```

```
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

CodeDeploy utilizza questo AppSpec file per copiare tutti i file nella `/tmp/WordPress` cartella sulla macchina di sviluppo nella `/var/www/html/WordPress` cartella sull'istanza Amazon EC2 di destinazione. Durante la distribuzione, CodeDeploy esegue gli script specificati come `root` nella `/var/www/html/WordPress/scripts` cartella sull'istanza Amazon EC2 di destinazione in occasione di eventi specifici durante il ciclo di vita della distribuzione, ad esempio e. **`BeforeInstall`** **`AfterInstall`** Se l'esecuzione di uno di questi script impiega più di 300 secondi (5 minuti), CodeDeploy interrompe la distribuzione e contrassegna la distribuzione come fallita.

Per ulteriori informazioni su queste impostazioni, consulta i CodeDeploy AppSpec riferimento al file.

> **Important**
>
> Le ubicazioni e i numeri di spazi tra ciascuna delle voci in questo file sono importanti. Se la spaziatura non è corretta, CodeDeploy genera un errore di cui potrebbe essere difficile eseguire il debug. Per ulteriori informazioni, consulta AppSpec Spaziatura tra i file.

Passaggio 3: carica l' WordPress applicazione su Amazon S3

Ora preparerai e caricherai i tuoi contenuti di origine in una posizione da cui CodeDeploy distribuirli. Le seguenti istruzioni mostrano come effettuare il provisioning di un bucket Amazon S3, preparare i file di revisione dell'applicazione per il bucket, raggruppare i file della revisione e quindi inserire la revisione nel bucket.

> **Note**
>
> Sebbene non sia trattato in questo tutorial, puoi CodeDeploy utilizzarlo per distribuire applicazioni dai repository alle istanze. GitHub Per ulteriori informazioni, consulta Integrazione con CodeDeploy GitHub.

Argomenti

- Esegui il provisioning di un bucket Amazon S3

- Prepara i file dell'applicazione per il bucket
- Raggruppate i file dell'applicazione in un unico file di archivio e inviate il file di archivio

Esegui il provisioning di un bucket Amazon S3

Crea un contenitore o un bucket di storage in Amazon S3 o usa un bucket esistente. Assicurati di poter caricare la revisione nel bucket e che le istanze Amazon EC2 utilizzate nelle distribuzioni possano scaricare la revisione dal bucket.

Puoi usare la AWS CLI console Amazon S3 o le API Amazon S3 per creare un bucket Amazon S3. Dopo aver creato il bucket, assicurati di concedere le autorizzazioni di accesso al bucket e al tuo account. AWS

> **Note**
>
> I nomi dei bucket devono essere univoci in Amazon S3 per AWS tutti gli account. Se non si è in grado di utilizzare **`amzn-s3-demo-bucket`**, provare un altro nome del bucket, ad esempio **`amzn-s3-demo-bucket`** seguito da un trattino e le iniziali o altri identificatori univoci. Assicurarsi quindi di sostituire il nome del bucket per **`amzn-s3-demo-bucket`** ovunque sia presente in questo tutorial.
> Il bucket Amazon S3 deve essere creato nella stessa AWS regione in cui vengono lanciate le istanze Amazon EC2 di destinazione. Ad esempio, se crei il bucket nella regione Stati Uniti orientali (Virginia settentrionale), le istanze Amazon EC2 di destinazione devono essere avviate nella regione Stati Uniti orientali (Virginia settentrionale).

Argomenti

- Per creare un bucket Amazon S3 (CLI)
- Per creare un bucket Amazon S3 (console)
- Concedi le autorizzazioni al bucket Amazon S3 e AWS account

Per creare un bucket Amazon S3 (CLI)

Chiama il mb comando per creare un bucket Amazon S3 denominato: **`amzn-s3-demo-bucket`**

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

Per creare un bucket Amazon S3 (console)

1. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Nella console Amazon S3, scegli Crea bucket.
3. Nella casella Nome bucket, digitare un nome per il bucket.
4. Nell'elenco delle regioni, scegli la regione di destinazione, quindi scegli Crea.

Concedi le autorizzazioni al bucket Amazon S3 e AWS account

È necessario disporre delle autorizzazioni per il caricamento nel bucket Amazon S3. Puoi specificare queste autorizzazioni tramite una policy sui bucket di Amazon S3. Ad esempio, nella seguente policy sui bucket di Amazon S3, l'utilizzo del carattere wildcard (*) consente `111122223333` all' AWS account di caricare file in qualsiasi directory del bucket Amazon S3 denominata: `amzn-s3-demo-bucket`

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

Per visualizzare l'ID del tuo AWS account, consulta Finding Your account ID. AWS

Ora è il momento giusto per verificare che il bucket Amazon S3 consenta le richieste di download da ogni istanza Amazon EC2 partecipante. Puoi specificarlo tramite una policy sui bucket di Amazon S3. Ad esempio, nella seguente policy sui bucket di Amazon S3, l'utilizzo del carattere wildcard (*) consente a qualsiasi istanza Amazon EC2 con un profilo di istanza IAM allegato contenente l'ARN di scaricare file `arn:aws:iam::444455556666:role/CodeDeployDemo` da qualsiasi directory del bucket Amazon S3 denominata: `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"
                ]
            }
        }
    ]
}
```

Per informazioni su come generare e allegare una bucket policy di Amazon S3, consulta Esempi di policy Bucket.

Per informazioni su come creare e allegare una policy IAM, consulta Working with policies.

Prepara i file dell'applicazione per il bucket

Assicurati che i file WordPress dell'applicazione, il AppSpec file e gli script siano organizzati sulla tua macchina di sviluppo in modo simile al seguente:

```
/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...)
```

Raggruppate i file dell'applicazione in un unico file di archivio e inviate il file di archivio

Raggruppa i file WordPress dell'applicazione e il AppSpec file in un file di archivio (noto come revisione dell'applicazione).

> **Note**
>
> Potrebbero essere addebitati dei costi per lo storage degli oggetti in un bucket e per il trasferimento delle revisioni dell'applicazione dentro e fuori un bucket. Per ulteriori informazioni, consulta Prezzi di Amazon S3.

1. Sul computer di sviluppo, passare alla cartella in cui sono archiviati i file:

   ```
   cd /tmp/WordPress
   ```

   > **Note**
   >
   > Se non si passa a questa cartella, il raggruppamento dei file verrà avviato nella cartella corrente. Ad esempio, se la cartella corrente è `/tmp` invece di `/tmp/WordPress`, il raggruppamento verrà quindi avviato con i file e le sottocartelle presenti nella cartella `tmp`, che può includere più di una sottocartella `WordPress`.

2. Chiamare il comando create-application per registrare una nuova applicazione denominata **`WordPress_App`**

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

3. Richiama il comando CodeDeploy push per raggruppare i file, caricare le revisioni su Amazon S3 e registrare le informazioni CodeDeploy sulla revisione caricata, il tutto in un'unica operazione.

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

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

Questo comando raggruppa i file dalla directory corrente (esclusi i file nascosti) in un unico file di archivio denominato**`WordPressApp.zip`**, carica la revisione nel **`amzn-s3-demo-bucket`** bucket e registra le informazioni sulla revisione caricata. CodeDeploy

Fase 4: Implementazione dell'applicazione WordPress

Ora distribuisci la revisione dell' WordPress applicazione di esempio che hai caricato su Amazon S3. Puoi utilizzare la console AWS CLI o la CodeDeploy console per distribuire la revisione e monitorare l'avanzamento della distribuzione. Una volta distribuita la revisione dell'applicazione, si possono controllare i risultati.

Argomenti

- Implementa la revisione dell'applicazione con CodeDeploy
- Monitora e risolvi i problemi della distribuzione
- Verifica la tua implementazione

Implementa la revisione dell'applicazione con CodeDeploy

Usa la console AWS CLI o la console per distribuire la revisione dell'applicazione.

Argomenti

- Per distribuire la revisione dell'applicazione (CLI)
- Per distribuire la revisione dell'applicazione (console)

Per distribuire la revisione dell'applicazione (CLI)

1. La distribuzione richiede un gruppo di distribuzione. Tuttavia, prima di creare il gruppo di distribuzione, è necessario un ARN del ruolo del servizio. Un ruolo di servizio è un ruolo IAM che concede al servizio l'autorizzazione ad agire per tuo conto. In questo caso, il ruolo di servizio consente di CodeDeploy accedere alle istanze Amazon EC2 per espandere (leggere) i relativi tag di istanza Amazon EC2.

Per creare un ruolo del servizio occorre avere già eseguito le istruzioni in Creare un ruolo di servizio (CLI). Per ottenere l'ARN del ruolo del servizio, consulta Ottieni il ruolo di servizio ARN (CLI).

2. Ora che hai il ruolo di servizio ARN, chiama il create-deployment-group comando per creare un gruppo di distribuzione denominato**`WordPress_DepGroup`**, associato all'applicazione denominata**`WordPress_App`**, utilizzando il tag Amazon EC2 denominato e la configurazione di distribuzione **`CodeDeployDemo`** denominata: **`CodeDeployDefault.OneAtATime`**

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

> **Note**
>
> Il comando create-deployment-group fornisce supporto per la creazione di trigger che determinano l'invio di notifiche Amazon SNS agli abbonati dell'argomento su eventi specifici nelle distribuzioni e nelle istanze. Il comando supporta anche opzioni per il rollback automatico delle distribuzioni e l'impostazione di allarmi per interrompere le distribuzioni quando vengono raggiunte le soglie di monitoraggio negli allarmi Amazon. CloudWatch I comandi per queste azioni non sono inclusi in questo tutorial.

3. Prima di creare una distribuzione, l' CodeDeploy agente deve essere installato sulle istanze del gruppo di distribuzione. È possibile installare l'agente dalla riga di comando con AWS Systems Manager tramite il seguente comando:

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

Questo comando crea un'associazione in Systems Manager State Manager che installerà l' CodeDeploy agente e quindi tenterà di aggiornarlo alle 2:00 ogni domenica mattina. Per ulteriori

informazioni sull' CodeDeploy agente, vedere Lavorare con l' CodeDeploy agente. Per ulteriori informazioni su Systems Manager, vedere What is AWS Systems Manager.

4. Chiamare ora il comando create-deployment per creare una distribuzione associata all'applicazione denominata **`WordPress_App`**, la configurazione della distribuzione denominata **`CodeDeployDefault.OneAtATime`** e il gruppo di distribuzione denominato **`WordPress_DepGroup`** utilizzando la revisione dell'applicazione denominata **`WordPressApp.zip`** nel bucket denominato **`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
```

Per distribuire la revisione dell'applicazione (console)

1. Prima di utilizzare la CodeDeploy console per distribuire la revisione dell'applicazione, è necessario un ARN per il ruolo di servizio. Un ruolo di servizio è un ruolo IAM che autorizza il servizio ad agire per conto dell'utente. In questo caso, il ruolo di servizio consente di CodeDeploy accedere alle istanze Amazon EC2 per espandere (leggere) i relativi tag di istanza Amazon EC2.

   Per creare un ruolo del servizio occorre avere già eseguito le istruzioni in Crea un ruolo di servizio (console). Per ottenere l'ARN del ruolo del servizio, consulta Ottieni il ruolo di servizio ARN (console).
2. Ora che hai l'ARN, usa la CodeDeploy console per distribuire la revisione dell'applicazione:

   Accedi a Console di gestione AWS e apri la console all' CodeDeploy indirizzo. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.
3. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
4. Nell'elenco delle applicazioni, scegli WordPress_App.
5. Nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione).

6. In Deployment group name (Nome del gruppo di distribuzione), immettere **`WordPress_DepGroup`**.
7. In Tipo di distribuzione, scegli In-placedistribuzione.
8. Nella configurazione dell'ambiente, seleziona le istanze Amazon EC2.
9. Nella configurazione dell'agente con AWS Systems Manager, mantieni le impostazioni predefinite.
10. In Key (Chiave), immettere **`Name`**.
11. In Valore, immetti **`CodeDeployDemo`**.

> **Note**
>
> Dopo aver digitato**`CodeDeployDemo`**, dovrebbe apparire un 1 in Istanze corrispondenti per confermare che è CodeDeploy stata trovata un'istanza Amazon EC2 corrispondente.

12. In Deployment configuration (Configurazione della distribuzione), scegliere CodeDeployDefault.OneAtATime.
13. In Service role ARN (ARN ruolo di servizio), scegliere l'ARN del ruolo di servizio, quindi scegliere Create deployment group (Crea gruppo di distribuzione).
14. Scegli Create deployment (Crea distribuzione).
15. In Deployment group (Gruppo di distribuzione) scegliere **`WordPress_DepGroup`**.
16. Accanto a Tipo di repository, seleziona La mia applicazione è archiviata in Amazon S3. In Luogo di revisione, inserisci la posizione della revisione dell' WordPress applicazione di esempio che hai precedentemente caricato su Amazon S3. Per ottenere la posizione:
    a. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
    b. Nell'elenco dei bucket, scegli amzn-s3-demo-bucket (o il nome del bucket in cui hai caricato la revisione dell'applicazione).
    c. Nell'elenco degli oggetti scegliere WordPressApp.zip.
    d. Nella scheda Panoramica copiare negli Appunti il valore del campo Link.

       Potrebbe essere simile al seguente:

       `https://s3.amazonaws.com/amzn-s3-demo-bucket/WordPressApp.zip`
    e. Torna alla CodeDeploy console e, nella posizione Revisione, incolla il valore del campo Link.

17. Se nell'elenco File type (Tipo di file) vene visualizzato un messaggio che informa che non è stato possibile trovare il tipo di file, scegliere .zip.
18. (Facoltativo) digitare un commento nella casella Deployment description (Descrizione distribuzione).
19. Espandere Deployment group overrides (Sostituzioni gruppo di distribuzione) e in Deployment configuration (Configurazione distribuzione) scegliere CodeDeployDefault.OneAtATime.
20. Selezionare Start deployment (Avvia distribuzione). Nella pagina Deployments (Distribuzioni) vengono visualizzate le informazioni relative alla nuova distribuzione creata.

Monitora e risolvi i problemi della distribuzione

Usa la console AWS CLI o la console per monitorare e risolvere i problemi della distribuzione.

Argomenti

- Per monitorare e risolvere problemi relativi alla distribuzione (CLI)
- Per monitorare e risolvere problemi relativi alla distribuzione (console)

Per monitorare e risolvere problemi relativi alla distribuzione (CLI)

1. Ottenere l'ID della distribuzione chiamando il comando list-deployments rispetto all'applicazione denominata **`WordPress_App`** e il gruppo di distribuzione denominato **`WordPress_DepGroup`**:

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

2. Chiamare il comando get-deployment con l'ID di distribuzione:

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

3. Il comando restituisce lo stato complessivo della distribuzione. In caso di esito positivo, il valore è `Succeeded`.

   Se lo stato generale è`Failed`, puoi chiamare comandi come list-deployment-instances e get-deployment-instance per la risoluzione dei problemi. Per ulteriori opzioni di risoluzione dei problemi, consulta Analisi dei file di log per individuare gli errori di distribuzione nelle istanze.

Per monitorare e risolvere problemi relativi alla distribuzione (console)

Nella pagina Distribuzioni della console, puoi monitorare lo stato della distribuzione nella colonna Stato. CodeDeploy

Per ottenere ulteriori informazioni sulla distribuzione, in particolare se il valore della colonna Status (Stato) è diverso da Succeeded (Riuscito):

1. Nella scheda Deployments (Distribuzioni) scegliere il nome della distribuzione. Se una distribuzione non riesce, viene visualizzato un messaggio che descrive il motivo dell'errore.
2. In Instance activity (Attività istanza) vengono visualizzate altre informazioni sulla distribuzione. Dopo un errore di distribuzione, potresti essere in grado di determinare su quali istanze Amazon EC2 e in quale fase l'implementazione non è riuscita.
3. Se si vuole procedere alla risoluzione del problema, è possibile avvalersi di una tecnica come quella descritta in View Instance Details. Puoi anche analizzare i file di log di distribuzione su un'istanza Amazon EC2. Per ulteriori informazioni, consulta Analisi dei file di log per individuare gli errori di distribuzione nelle istanze.

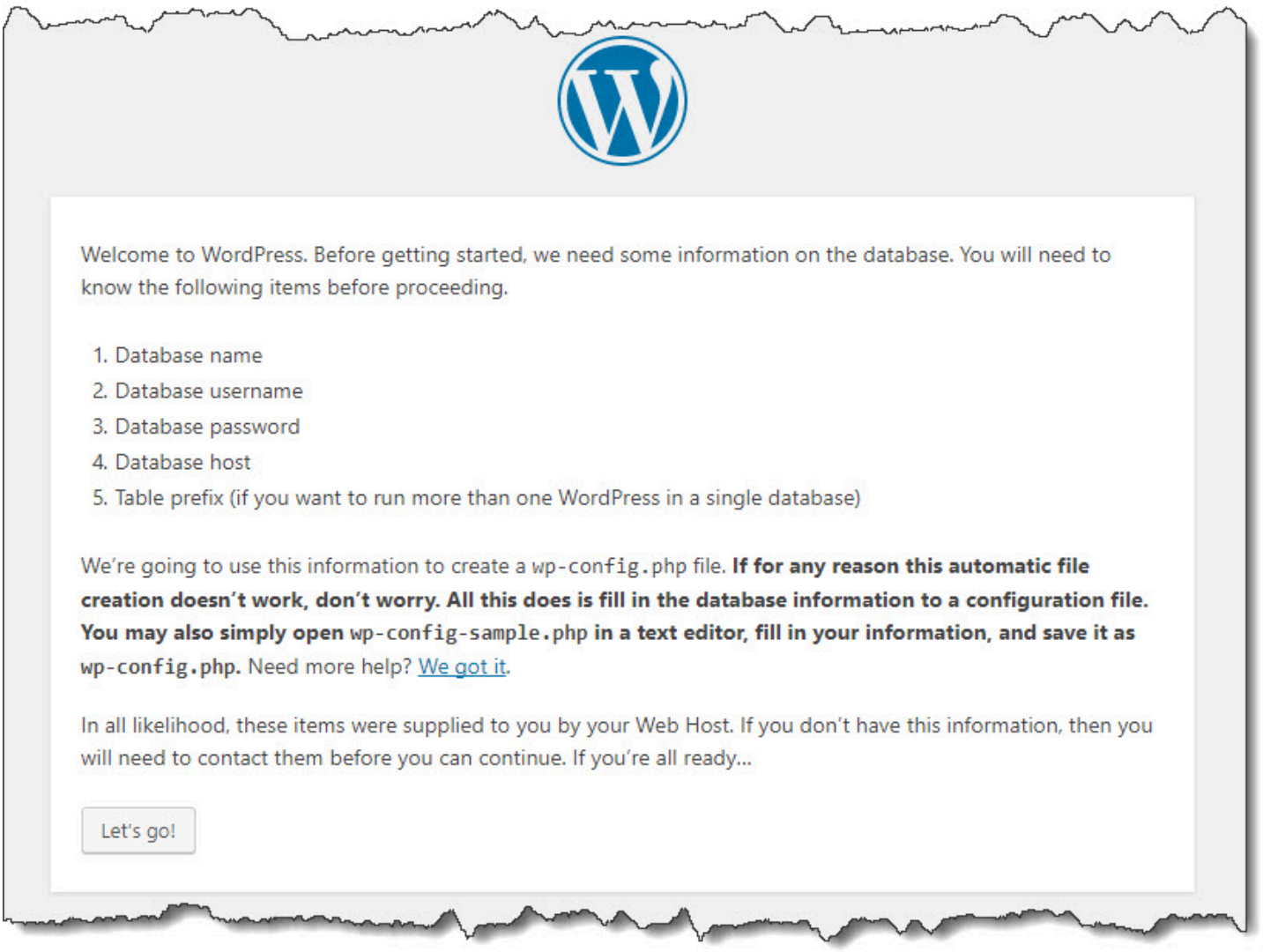
Verifica la tua implementazione

Una volta completata la distribuzione, verifica che l' WordPress installazione funzioni. Utilizza l'indirizzo DNS pubblico dell'istanza Amazon EC2, seguito `/WordPress` da, per visualizzare il tuo sito in un browser Web. (Per ottenere il valore DNS pubblico, nella console Amazon EC2, scegli l'istanza Amazon EC2 e, nella scheda Descrizione, cerca il valore di Public DNS.)

Ad esempio, se l'indirizzo DNS pubblico della tua istanza Amazon EC2 **`ec2-01-234-567-890.compute-1.amazonaws.com`** è, utilizzerai il seguente URL:

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

Quando visualizzi il sito nel tuo browser, dovresti vedere una pagina di WordPress benvenuto simile alla seguente:



Se alla tua istanza Amazon EC2 non è stata aggiunta una regola di ingresso HTTP al relativo gruppo di sicurezza, la pagina di WordPress benvenuto non viene visualizzata. Se vedi un messaggio che indica che il server remoto non risponde, assicurati che il gruppo di sicurezza per la tua istanza Amazon EC2 abbia la regola in entrata. Per ulteriori informazioni, consulta Aggiungi una regola in entrata che consenta il traffico HTTP verso la tua istanza Amazon Linux o RHEL Amazon EC2.

Fase 5: Aggiornare e ridistribuire l'applicazione WordPress

Ora che hai distribuito correttamente la revisione dell'applicazione, aggiorna il WordPress codice sulla macchina di sviluppo e CodeDeploy usalo per ridistribuire il sito. Successivamente, dovresti vedere le modifiche al codice sull'istanza Amazon EC2.

Argomenti

- Configura il sito WordPress
- Modifica il sito
- Ridistribuisci il sito

Configura il sito WordPress

Per vedere gli effetti della modifica del codice, completa la configurazione del WordPress sito in modo da avere un'installazione completamente funzionante.

1. Digitare l'URL del sito nel browser Web. L'URL è l'indirizzo DNS pubblico dell'istanza Amazon EC2 più `/WordPress` un'estensione. Per questo WordPress sito di esempio (e ad esempio l'indirizzo DNS pubblico dell'istanza Amazon EC2), l'URL è. **`http://ec2-01-234-567-890.compute-1.amazonaws.com/WordPress`**
2. Se non hai ancora configurato il sito, viene visualizzata la pagina di benvenuto WordPress predefinita. Selezionare Let's go! (Iniziamo!).
3. Per utilizzare l'impostazione predefinita del database MySQL, nella pagina di configurazione del database digitare i seguenti valori:

   - Database Name (Nome del database): **`test`**
   - User Name: **`root`**
   - Password: lasciare vuoto.
   - Database Host: **`localhost`**
   - Table Prefix: **`wp_`**

   Selezionare Submit (Invia) per configurare il database.
4. Continuare la configurazione del sito. Nella pagina di benvenuto, inserisci i valori che desideri e scegli Installa WordPress. Al termine dell'installazione, è possibile accedere al pannello di controllo.

> **Important**
>
> Durante la distribuzione dell' WordPress applicazione, **`change_permissions.sh`** lo script ha aggiornato le autorizzazioni della `/tmp/WordPress` cartella in modo che chiunque possa scrivervi. In questa fase è possibile eseguire il comando seguente per limitare le autorizzazioni in modo che solo il proprietario possa scrivere nella cartella:

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

Modifica il sito

Per modificare il WordPress sito, vai alla cartella dell'applicazione sulla tua macchina di sviluppo:

```
cd /tmp/WordPress
```

Per modificare alcuni dei colori del sito, nel file `wp-content/themes/twentyfifteen/style.css` utilizzare un editor di testo o sed per modificare `#fff` in `#768331`.

In Linux o altri sistemi con GNU sed, utilizzare:

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

In macOS, Unix o altri sistemi con BSD sed, utilizzare:

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

Ridistribuisci il sito

Ora che hai modificato il codice del sito, usa Amazon S3 e ridistribuisci CodeDeploy il sito.

Raggruppa e carica le modifiche su Amazon S3, come descritto in. Raggruppate i file dell'applicazione in un unico file di archivio e inviate il file di archivio (Nel seguire queste istruzioni, ricordare che non è necessario creare un'applicazione). Assegnare alla nuova revisione la stessa chiave di prima (**`WordPressApp.zip`**). Caricalo nello stesso bucket Amazon S3 che hai creato in precedenza (ad esempio,). **`amzn-s3-demo-bucket`**

Usa la AWS CLI, la CodeDeploy console o le CodeDeploy API per ridistribuire il sito.

Argomenti

- Per ridistribuire il sito (CLI)
- Per ridistribuire il sito (console)

Per ridistribuire il sito (CLI)

Chiamare il comando create-deployment per creare una distribuzione basata sulla revisione appena caricata. Utilizzare l'applicazione denominata **WordPress_App**, la configurazione della distribuzione denominata **CodeDeployDefault.OneAtATime**, il gruppo di distribuzione denominato **WordPress_DepGroup** e la revisione denominata **WordPressApp.zip** nel bucket denominato **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
```

È possibile controllare lo stato della distribuzione, come descritto in Monitora e risolvi i problemi della distribuzione.

Dopo CodeDeploy aver ridistribuito il sito, visita nuovamente il sito nel browser Web per verificare che i colori siano stati modificati. (Potrebbe essere necessario aggiornare il browser). Se i colori sono stati modificati, la modifica e la ridistribuzione del sito sono riuscite.

Per ridistribuire il sito (console)

1. Accedi a Console di gestione AWS e apri la console all' CodeDeploy indirizzo. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Nell'elenco delle applicazioni, scegli WordPress_App.
4. Nella scheda Deployment groups (Gruppi di distribuzione), scegliere **WordPress_DepGroup**.
5. Scegli Create deployment (Crea distribuzione).
6. Nella pagina Create deployment (Crea distribuzione):

   a. In Deployment group (Gruppo di distribuzione), scegliere **WordPress_DepGroup**.

b. Nell'area Tipo di archivio, scegli La mia applicazione è archiviata in Amazon S3, quindi copia il link Amazon S3 della tua revisione nella casella Posizione di revisione. Per individuare il valore del link:

   i. In una scheda del browser separata:

      Accedi a Console di gestione AWS e apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/

      Naviga e apri amzn-s3-demo-bucket, quindi scegli la tua revisione,. **`WordPressApp.zip`**

   ii. Se il riquadro Proprietà non è visibile nella console Amazon S3, scegli il pulsante Proprietà.

   iii. Nel riquadro Proprietà, copia il valore del campo Link nella casella Posizione di revisione della CodeDeploy console.

c. Se viene visualizzato un messaggio che informa che non è stato possibile individuare il tipo di file, scegliere .zip.

d. Lasciare vuota la casella Deployment description (Descrizione distribuzione).

e. Espandere Deployment group overrides (Sovrascritture gruppo di distribuzione) e in Deployment configuration (Configurazione distribuzione) scegliere CodeDeployDefault.OneAtATime.

f. Selezionare Start deployment (Avvia distribuzione). Nella pagina Deployments (Distribuzioni) vengono visualizzate le informazioni relative alla nuova distribuzione creata.

g. È possibile controllare lo stato della distribuzione, come descritto in Monitora e risolvi i problemi della distribuzione.

   Dopo CodeDeploy aver ridistribuito il sito, rivisita il sito nel tuo browser web per verificare che i colori siano stati cambiati. (Potrebbe essere necessario aggiornare il browser). Se i colori sono stati modificati, la modifica e la ridistribuzione del sito sono riuscite.

Passaggio 6: Pulisci WordPress l'applicazione e le risorse correlate

Ora hai effettuato con successo un aggiornamento del WordPress codice e ridistribuito il sito. Per evitare addebiti continuativi per risorse create per questo tutorial, devi eliminare:

- Qualsiasi CloudFormation stack (o termina qualsiasi istanza Amazon EC2, se le hai create all'esterno). CloudFormation
- Qualsiasi bucket Amazon S3.
- L'applicazione `WordPress_App` in CodeDeploy.
- L'associazione AWS Systems Manager State Manager dell' CodeDeploy agente.

Puoi utilizzare Amazon S3 CloudFormation, Amazon EC2 CodeDeploy e console o le API per eseguire AWS la pulizia. AWS CLI

Argomenti

- Per cancellare le risorse (CLI)
- Per cancellare le risorse (console)
- Fasi successive

Per cancellare le risorse (CLI)

1. Se hai usato il nostro CloudFormation modello per questo tutorial, chiama il delete-stack comando sullo stack denominato. **`CodeDeployDemoStack`** Ciò interromperà tutte le istanze Amazon EC2 associate ed eliminerà tutti i ruoli IAM associati allo stack creato:

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

2. Per eliminare il bucket Amazon S3, chiama il rm comando con lo --recursive switch accanto al bucket denominato. **`amzn-s3-demo-bucket`** Il bucket e tutti gli oggetti in esso contenuti saranno quindi eliminati:

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

3. Per eliminare l'applicazione `WordPress_App`, chiamare il comando delete-application. In questo modo verranno inoltre eliminati tutti i record del gruppo di distribuzione associato e i record di distribuzione per l'applicazione:

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

4. Per eliminare l'associazione Systems Manager State Manager, chiamare il delete-association comando.

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

È possibile ottenerla *association-id* chiamando il describe-association comando.

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

Se non hai utilizzato lo CloudFormation stack per questo tutorial, chiama il terminate-instances comando per terminare tutte le istanze Amazon EC2 che hai creato manualmente. Fornisci l'ID dell'istanza Amazon EC2 da terminare:

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

Per cancellare le risorse (console)

Se hai utilizzato il nostro CloudFormation modello per questo tutorial, elimina lo stack associato CloudFormation .

1. Accedi a Console di gestione AWS e apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Nella casella Filtro, digita il nome CloudFormation dello stack creato in precedenza (ad esempio,**CodeDeployDemoStack**).
3. Selezionare la casella accanto al nome dello stack. Nel menu Actions (Operazioni), scegli Delete Stack (Elimina stack).

   CloudFormation elimina lo stack, termina tutte le istanze Amazon EC2 associate ed elimina tutti i ruoli IAM associati.

Per terminare le istanze Amazon EC2 che hai creato al di fuori di uno stack: CloudFormation

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Nell'elenco INSTANCES, scegliere Instances (Istanze).
3. Nella casella di ricerca, digita il nome dell'istanza Amazon EC2 che desideri terminare (ad esempio,**CodeDeployDemo**), quindi premi Invio.

4. Scegli il nome dell'istanza Amazon EC2.
5. Nel menu Actions (Operazioni), passare a Instance State (Stato istanza), quindi selezionare Terminate (Termina). Quando richiesto, scegliere Yes, Terminate (Sì, termina).

Ripeti queste fasi per ogni istanza.

Per eliminare il bucket Amazon S3:

1. Accedi Console di gestione AWS e apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Nell'elenco dei bucket, cerca e scegli il nome del bucket Amazon S3 che hai creato in precedenza (ad esempio,). **`amzn-s3-demo-bucket`**
3. Per eliminare un bucket, occorre innanzitutto rimuoverne i contenuti. Selezionare tutti i file nel bucket, ad esempio **`WordPressApp.zip`**. Dal menu Actions (Operazioni), scegliere Delete (Elimina). Quando viene richiesto di confermare l'eliminazione, scegliere OK (OK).
4. Una volta svuotato, il bucket può essere eliminato. Nell'elenco dei bucket, selezionare la riga del bucket d'interesse (non il suo nome). Selezionare Delete bucket (Elimina bucket) e, quando viene richiesto di confermare, selezionare OK (OK).

Per eliminare l'applicazione da: `WordPress_App` CodeDeploy

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Nell'elenco delle applicazioni, scegli WordPress_App.
4. Nella pagina Application details (Dettagli applicazione), scegliere Delete application (Elimina applicazione).
5. Quando richiesto, digitare il nome dell'applicazione per confermare che si desidera eliminarla, quindi scegliere Delete (Elimina).

Per eliminare l'associazione Systems Manager State Manager:

1. Aprire la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager.
2. Nel riquadro di navigazione, seleziona State Manager.
3. Scegli l'associazione creata e seleziona Elimina.

Fasi successive

Se sei arrivato qui, complimenti. Hai completato una distribuzione CodeDeploy e quindi aggiornato e ridistribuito il codice del sito.

Tutorial: distribuisci un messaggio «ciao, mondo!» applicazione con CodeDeploy (Windows Server)

In questo tutorial, distribuisci una singola pagina Web su una singola istanza Amazon EC2 di Windows Server che esegue Internet Information Services (IIS) come server Web. Questa pagina Web mostra un semplice messaggio «Hello, World!» messaggio.

Non hai trovato ciò che cerchi?

- Per fare pratica con la distribuzione su un'istanza Amazon EC2 di Amazon Linux o Red Hat Enterprise Linux (RHEL), consulta. Tutorial: Esegui WordPress la distribuzione su un'istanza Amazon EC2 (Amazon Linux o Red Hat Enterprise Linux e Linux, macOS o Unix)
- Per esercitarti nella distribuzione su un'istanza locale, invece, consulta Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux).

Le fasi di questo tutorial presuppongono l'utilizzo di Windows. Sebbene sia possibile completare la maggior parte di questi passaggi su un computer locale che esegue Linux, macOS o Unix, è necessario adattare quelli che coprono i percorsi di Windows-based directory come. `c:\temp` Inoltre, se desideri connetterti all'istanza Amazon EC2, hai bisogno di un'applicazione client in grado di connettersi tramite Remote Desktop Protocol (RDP) all'istanza Amazon EC2 che esegue Windows Server. (Windows include, per impostazione predefinita, un'applicazione client di connessione tramite RDP.)

Prima di iniziare questo tutorial, devi completare i prerequisiti di, tra cui la configurazione dell'utenteGuida introduttiva con CodeDeploy, l'installazione o l'aggiornamento e la creazione di un profilo di istanza IAM e di un ruolo di servizio. AWS CLI

Argomenti

- Fase 1: Avvio di un'istanza Amazon EC2 per Windows Server
- Fase 2: configura il contenuto sorgente da distribuire sull'istanza Amazon EC2 di Windows Server
- Passaggio 3: Carica il tuo «ciao, mondo!» applicazione per Amazon S3
- Fase 4: Implementa la tua applicazione Hello World
- Passaggio 5: aggiorna e ridistribuisci il tuo «ciao, mondo!» applicazione
- Passaggio 6: ripulisci il tuo «ciao mondo!» applicazione e risorse correlate

Fase 1: Avvio di un'istanza Amazon EC2 per Windows Server

Per distribuire l'applicazione Hello World con CodeDeploy, è necessaria un'istanza Amazon EC2 con Windows Server.

Segui le istruzioni in Crea un'istanza Amazon EC2 per CodeDeploy. Quando sei pronto per assegnare un tag di istanza Amazon EC2 all'istanza, assicurati di specificare la chiave del tag e il valore **`Name`** del tag di. **`CodeDeployDemo`** Se viene specificata una chiave o un valore del tag, le istruzioni in Fase 4: Implementa la tua applicazione Hello World possono portare a risultati inaspettati.

Dopo aver avviato l'istanza Amazon EC2, torna a questa pagina e passa alla sezione successiva. Non continuare su Crea un'applicazione con CodeDeploy come fase successiva.

Connect alla tua istanza Amazon EC2

Dopo l'avvio dell'istanza Amazon EC2, segui queste istruzioni per esercitarti a connetterti ad essa.

> **Note**
>
> In queste istruzioni, supponiamo che siano in esecuzione Windows e l'applicazione client Windows Desktop Connection. Per informazioni, consulta Connessione all'istanza Windows tramite RDP. Potrebbe essere necessario adeguare queste istruzioni agli altri sistemi operativi o alle applicazioni client di connessione tramite RDP.

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Nel riquadro di navigazione, in Istanze scegli Istanze.
3. Naviga e scegli la tua istanza di Windows Server nell'elenco.
4. Scegli Connetti.
5. Scegli Richiedi la password e seleziona Scegli file.
6. Cerca e scegli il file della coppia di chiavi dell'istanza Amazon EC2 associato all'istanza Amazon EC2 di Windows Server, quindi scegli Apri.
7. Selezionare Decrypt Password (Decifra password). Annotare la password che viene visualizzata. Servirà nella fase 10.
8. Selezionare Download Remote Desktop File (Scarica file per il desktop remoto) e aprire il file.
9. Continuare nel caso in cui venga richiesta la connessione anche se il publisher della connessione remota non può essere identificato.
10. Digitare la password annotata durante la fase 7, quindi continuare. Se l'applicazione client della connessione tramite RDP richiede un nome utente, digitare **`Administrator`**.
11. Continuare nel caso in cui venga richiesta la connessione anche se l'identità del computer remoto non può essere verificata.
12. Dopo la connessione, viene visualizzato il desktop dell'istanza Amazon EC2 che esegue Windows Server.
13. Ora puoi disconnetterti dall'istanza Amazon EC2.

> **Warning**
>
> Non arrestare o terminare l'istanza. In caso contrario, non CodeDeploy è possibile eseguire la distribuzione su di essa.

Aggiungi una regola in entrata che consenta il traffico HTTP verso la tua istanza Amazon EC2 di Windows Server

Il passaggio successivo conferma che la tua istanza Amazon EC2 ha una porta HTTP aperta in modo da poter vedere la pagina Web distribuita sulla tua istanza Amazon EC2 Windows Server in un browser.

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Scegli Istanze, quindi scegli la tua istanza.
3. Nella scheda Descrizione, in Gruppi di sicurezza, scegli Visualizza regole in entrata.

   Dovresti vedere un elenco di regole nel tuo gruppo di sicurezza come il seguente:

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

4. In Gruppi di sicurezza, scegli il gruppo di sicurezza per la tua istanza Amazon EC2. Potrebbe essere denominato **`launch-wizard-N`**. Il nome ***N*** è un numero assegnato al proprio gruppo di sicurezza quando è stata creata l'istanza.

   Seleziona la scheda Inbound (In entrata). Se il gruppo di sicurezza per la tua istanza è configurato correttamente, dovresti vedere una regola con i seguenti valori:

   - Type (Tipo): HTTP
   - Protocollo: TCP
   - Port Range (Intervallo porte): 80
   - Fonte: 0.0.0. 0/0
5. Se non vedi una regola con questi valori, usa le procedure in Aggiungere regole a un gruppo di sicurezza per aggiungerle a una nuova regola di sicurezza.

Fase 2: configura il contenuto sorgente da distribuire sull'istanza Amazon EC2 di Windows Server

Ora è il momento di configurare il contenuto sorgente della tua applicazione in modo da avere qualcosa da distribuire sull'istanza Amazon EC2. Per questo tutorial, distribuirai una singola pagina Web sull'istanza Amazon EC2 che esegue Windows Server, che eseguirà Internet Information Services (IIS) come server Web. Questa pagina Web mostrerà un semplice messaggio «Hello, World!» messaggio.

Argomenti

- Crea la pagina web

- Crea uno script per eseguire l'applicazione
- Aggiungi un file con le specifiche dell'applicazione

Crea la pagina web

1. Creare una sottodirectory (sottocartella) denominata `HelloWorldApp` nella propria cartella `c:\temp` e quindi passare a questa cartella.

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

> **Note**
>
> Non è necessario utilizzare la posizione `c:\temp` o il nome della sottocartella di `HelloWorldApp`. Se si utilizza una posizione o un nome della sottocartella diverso, assicurarsi di usarlo durante tutto il tutorial.

2. Utilizzare un editor di testo per creare un file all'interno della cartella. Assegnare un nome al file `index.html`.

```
notepad index.html
```

3. Aggiungere il codice HTML seguente al file, quindi salvare il file.

```
<!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>
```

Crea uno script per eseguire l'applicazione

Successivamente, creerai uno script che CodeDeploy verrà utilizzato per configurare il server Web sull'istanza Amazon EC2 di destinazione.

1. Nella stessa sottocartella in cui viene salvato il file `index.html`, utilizzare un editor di testo per creare un altro file. Assegnare un nome al file `before-install.bat`.

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

2. Aggiungere lo script batch seguente al file, quindi salvare il file.

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

Aggiungi un file con le specifiche dell'applicazione

Successivamente, aggiungerete un file di specifiche dell'applicazione (AppSpec file) oltre alla pagina Web e al file di script batch. Il AppSpec file è un file in formato YAML utilizzato da: CodeDeploy

- Mappare i file sorgente nella revisione dell'applicazione nelle loro destinazioni sull'istanza.
- Specificare gli script da eseguire sull'istanza durante la distribuzione.

Il AppSpec file deve avere un nome. `appspec.yml` Deve essere inserito nella cartella root del codice sorgente dell'applicazione.

1. Nella stessa sottocartella in cui vengono salvati i file `index.html` e `before-install.bat`, utilizzare un editor di testo per creare un altro file. Assegnare un nome al file `appspec.yml`.

```
notepad appspec.yml
```

2. Aggiungere il codice YAML seguente al file, quindi salvare il file.

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

CodeDeploy utilizzerà questo AppSpec file per copiare il `index.html` file nella cartella principale del codice sorgente dell'applicazione nella `c:\inetpub\wwwroot` cartella sull'istanza Amazon EC2 di destinazione. Durante la distribuzione, CodeDeploy eseguirà lo script `before-install.bat` batch sull'istanza Amazon EC2 di destinazione durante l'evento del ciclo di vita della **`BeforeInstall`** distribuzione. Se l'esecuzione di questo script impiega più di 900 secondi (15 minuti), CodeDeploy interromperà la distribuzione e contrassegnerà la distribuzione sull'istanza Amazon EC2 come fallita.

Per ulteriori informazioni su queste impostazioni, consulta i CodeDeploy AppSpec riferimento al file.

> **Important**
>
> Le ubicazioni e i numeri di spazi tra ciascuna delle voci in questo file sono importanti. Se la spaziatura non è corretta, CodeDeploy verrà generato un errore di cui potrebbe essere difficile eseguire il debug. Per ulteriori informazioni, consulta AppSpec Spaziatura tra i file.

Passaggio 3: Carica il tuo «ciao, mondo!» applicazione per Amazon S3

Ora preparerai e caricherai i tuoi contenuti di origine in una posizione da cui CodeDeploy distribuirli. Le seguenti istruzioni mostrano come effettuare il provisioning di un bucket Amazon S3, preparare i file di revisione dell'applicazione per il bucket, raggruppare i file della revisione e quindi inserire la revisione nel bucket.

> **Note**
>
> Sebbene non sia trattato in questo tutorial, puoi CodeDeploy utilizzarlo per distribuire applicazioni dai repository alle istanze. GitHub Per ulteriori informazioni, consulta Integrazione con CodeDeploy GitHub.

Argomenti

- Esegui il provisioning di un bucket Amazon S3
- Prepara i file dell'applicazione per il bucket
- Raggruppa i file dell'applicazione in un unico file di archivio e invia il file di archivio

Esegui il provisioning di un bucket Amazon S3

Crea un contenitore o un bucket di storage in Amazon S3 o usa un bucket esistente. Assicurati di poter caricare la revisione nel bucket e che le istanze Amazon EC2 utilizzate nelle distribuzioni possano scaricare la revisione dal bucket.

Puoi usare la AWS CLI console Amazon S3 o le API Amazon S3 per creare un bucket Amazon S3. Dopo la creazione del bucket, il bucket e l'utente CodeDeploy devono avere l'accesso alle autorizzazioni.

> **Note**
>
> I nomi dei bucket devono essere univoci in Amazon S3 per AWS tutti gli account. Se non si è in grado di utilizzare **`amzn-s3-demo-bucket`**, provare un altro nome del bucket, ad esempio **`amzn-s3-demo-bucket`** seguito da un trattino e le iniziali o altri identificatori univoci. Assicurarsi quindi di sostituire il nome del bucket per **`amzn-s3-demo-bucket`** ovunque sia presente in questo tutorial.
> Il bucket Amazon S3 deve essere creato nella stessa AWS regione in cui vengono lanciate le istanze Amazon EC2 di destinazione. Ad esempio, se crei il bucket nella regione Stati Uniti orientali (Virginia settentrionale), le istanze Amazon EC2 di destinazione devono essere avviate nella regione Stati Uniti orientali (Virginia settentrionale).

Argomenti

- Per creare un bucket Amazon S3 (CLI)
- Per creare un bucket Amazon S3 (console)
- Concedi le autorizzazioni al bucket Amazon S3 e al AWS account

Per creare un bucket Amazon S3 (CLI)

Chiama il mb comando per creare un bucket Amazon S3 denominato: **`amzn-s3-demo-bucket`**

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

Per creare un bucket Amazon S3 (console)

1. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Nella console Amazon S3, scegli Crea bucket.
3. Nella casella Nome bucket, digitare un nome per il bucket.
4. Nell'elenco delle regioni, scegli la regione di destinazione, quindi scegli Crea.

Concedi le autorizzazioni al bucket Amazon S3 e al AWS account

È necessario disporre delle autorizzazioni per il caricamento nel bucket Amazon S3. Puoi specificare queste autorizzazioni tramite una policy sui bucket di Amazon S3. Ad esempio, nella seguente policy sui bucket di Amazon S3, l'utilizzo del carattere wildcard (*) consente `111122223333` all' AWS account di caricare file in qualsiasi directory del bucket Amazon S3 denominata: `amzn-s3-demo-bucket`

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

```
            }
        }
    ]
}
```

Per visualizzare l'ID del tuo AWS account, consulta Finding Your account ID. AWS

Ora è il momento giusto per verificare che il bucket Amazon S3 consenta le richieste di download da ogni istanza Amazon EC2 partecipante. Puoi specificarlo tramite una policy sui bucket di Amazon S3. Ad esempio, nella seguente policy sui bucket di Amazon S3, l'utilizzo del carattere wildcard (*) consente a qualsiasi istanza Amazon EC2 con un profilo di istanza IAM allegato contenente l'ARN di scaricare file `arn:aws:iam::444455556666:role/CodeDeployDemo` da qualsiasi directory del bucket Amazon S3 denominata: `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"
                ]
            }
        }
    ]
}
```

Per informazioni su come generare e allegare una bucket policy di Amazon S3, consulta Esempi di policy Bucket.

L'utente CodeDeploy amministrativo in cui hai creato Fase 1: Configurazione deve inoltre disporre dell'autorizzazione per caricare la revisione nel bucket Amazon S3. Un modo per specificarlo è tramite una policy IAM, che aggiungi al set di autorizzazioni dell'utente, o a un ruolo IAM (che consenti all'utente di assumere). La seguente policy IAM consente all'utente di caricare le revisioni in qualsiasi punto del bucket Amazon S3 denominato: `amzn-s3-demo-bucket`

JSON

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

Per informazioni su come creare una policy IAM, consulta Creating IAM policies nella IAM User Guide. Per informazioni sull'aggiunta di una policy a un set di autorizzazioni, consulta Create a permission set nella Guida per l'Centro identità AWS IAM utente.

Prepara i file dell'applicazione per il bucket

Assicurati che la pagina web, il AppSpec file e lo script siano organizzati sulla tua macchina di sviluppo in questo modo:

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

Raggruppa i file dell'applicazione in un unico file di archivio e invia il file di archivio

Raggruppare i file presenti in un file di archivio (noti come revisione dell'applicazione).

> **Note**
>
> Potrebbero essere addebitati dei costi per lo storage degli oggetti in un bucket e per il trasferimento delle revisioni dell'applicazione dentro e fuori un bucket. Per ulteriori informazioni, consulta Prezzi di Amazon S3.

1. Sul computer di sviluppo, passare alla cartella in cui sono archiviati i file:

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

> **Note**
>
> Se non si passa a questa cartella, il raggruppamento dei file verrà avviato nella cartella corrente. Ad esempio, se la cartella corrente è `c:\temp` invece di `c:\temp\HelloWorldApp`, il raggruppamento verrà avviato con i file e le sottocartelle presenti nella cartella `c:\temp`, che può includere più di una sottocartella `HelloWorldApp`.

2. Chiamate il create-application comando per registrare una nuova applicazione denominata **`HelloWorld_App`** con CodeDeploy:

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

3. Richiama il comando CodeDeploy push per raggruppare i file, caricare le revisioni su Amazon S3 e registrare le informazioni CodeDeploy sulla revisione caricata, il tutto in un'unica operazione.

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

Questo comando raggruppa i file dalla directory corrente (esclusi i file nascosti) in un unico file di archivio denominato`HelloWorld_App.zip`, carica la revisione nel **`amzn-s3-demo-bucket`** bucket e registra le informazioni sulla revisione caricata. CodeDeploy

Fase 4: Implementa la tua applicazione Hello World

Ora distribuisci la revisione di esempio dell'applicazione Hello World che hai caricato su Amazon S3. Utilizzi la console AWS CLI o la CodeDeploy console per distribuire la revisione e monitorare l'avanzamento della distribuzione. Una volta distribuita la revisione dell'applicazione, si possono controllare i risultati.

Argomenti

- Implementa la revisione dell'applicazione con CodeDeploy
- Monitora e risolvi i problemi della distribuzione
- Verifica la tua implementazione

Implementa la revisione dell'applicazione con CodeDeploy

Puoi distribuire la tua applicazione utilizzando la CLI o la console.

Argomenti

- Per distribuire la revisione dell'applicazione (CLI)
- Per distribuire la revisione dell'applicazione (console)

Per distribuire la revisione dell'applicazione (CLI)

1. La distribuzione richiede innanzitutto un gruppo di distribuzione. Tuttavia, prima di creare il gruppo di distribuzione, è necessario un ARN del ruolo del servizio. Un ruolo di servizio è un ruolo IAM che concede al servizio l'autorizzazione ad agire per tuo conto. In questo caso, il ruolo di servizio consente di CodeDeploy accedere alle istanze Amazon EC2 per espandere (leggere) i relativi tag di istanza Amazon EC2.

   Per creare un ruolo del servizio occorre avere già eseguito le istruzioni in Creare un ruolo di servizio (CLI). Per ottenere l'ARN del ruolo del servizio, consulta Ottieni il ruolo di servizio ARN (CLI).

2. Ora che hai l'ARN, chiama il create-deployment-group comando per creare un gruppo di distribuzione denominato**`HelloWorld_DepGroup`**, associato all'applicazione denominata**`HelloWorld_App`**, utilizzando il tag di istanza Amazon EC2 denominato e la configurazione **`CodeDeployDefault.OneAtATime`** di distribuzione **`CodeDeployDemo`** denominata, con il ruolo di servizio 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**
   >
   > Il comando create-deployment-group fornisce supporto per la creazione di trigger che determinano l'invio di notifiche Amazon SNS agli abbonati dell'argomento su eventi specifici nelle distribuzioni e nelle istanze. Il comando supporta anche opzioni per il rollback automatico delle distribuzioni e l'impostazione di allarmi per interrompere le

distribuzioni quando vengono raggiunte le soglie di monitoraggio negli allarmi Amazon. CloudWatch I comandi per queste azioni non sono inclusi in questo tutorial.

3. Prima di creare una distribuzione, l' CodeDeploy agente deve essere installato sulle istanze del gruppo di distribuzione. È possibile installare l'agente dalla riga di comando con AWS Systems Manager tramite il seguente comando:

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

   Questo comando crea un'associazione in Systems Manager State Manager che installerà l' CodeDeploy agente e quindi tenterà di aggiornarlo alle 2:00 ogni domenica mattina. Per ulteriori informazioni sull' CodeDeploy agente, vedere Lavorare con l' CodeDeploy agente. Per ulteriori informazioni su Systems Manager, vedere What is AWS Systems Manager.

4. Chiamare ora il comando create-deployment per creare una distribuzione associata all'applicazione denominata **`HelloWorld_App`**, la configurazione della distribuzione denominata **`CodeDeployDefault.OneAtATime`** e il gruppo di distribuzione denominato **`HelloWorld_DepGroup`** utilizzando la revisione dell'applicazione denominata **`HelloWorld_App.zip`** nel bucket denominato **`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
```

Per distribuire la revisione dell'applicazione (console)

1. Prima di utilizzare la CodeDeploy console per distribuire la revisione dell'applicazione, è necessario un ARN per il ruolo di servizio. Un ruolo di servizio è un ruolo IAM che autorizza il servizio ad agire per conto dell'utente. In questo caso, il ruolo di servizio consente di CodeDeploy accedere alle istanze Amazon EC2 per espandere (leggere) i relativi tag di istanza Amazon EC2.

   Per creare un ruolo del servizio occorre avere già eseguito le istruzioni in Crea un ruolo di servizio (console). Per ottenere l'ARN del ruolo del servizio, consulta Ottieni il ruolo di servizio ARN (console).

2. Ora che hai l'ARN, puoi usare la CodeDeploy console per distribuire la revisione dell'applicazione.

   Accedi a Console di gestione AWS e apri la console all' CodeDeploy indirizzo. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

3. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
4. Scegli HelloWorld_App.
5. Nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione).
6. In Deployment group name (Nome del gruppo di distribuzione), immettere **`HelloWorld_DepGroup`**.
7. In Service Role (Ruolo del servizio), scegliere il nome del ruolo del servizio.
8. In Tipo di distribuzione, scegli In-place.
9. Nella configurazione dell'ambiente, seleziona le istanze Amazon EC2.
10. Nella configurazione dell'agente con AWS Systems Manager, mantieni le impostazioni predefinite.
11. In Key (Chiave), immettere **`Name`**.
12. In Valore, immetti **`CodeDeployDemo`**.
13. In Deployment configuration (Configurazione della distribuzione), scegliere CodeDeployDefault.OneAtATime.
14. In Load Balancer (Sistema di bilanciamento del carico), deselezionare Enable load balancing (Abilita il bilanciamento del carico).
15. Scegliere Create deployment group (Crea gruppo di distribuzione).
16. Scegli Create deployment (Crea distribuzione).
17. Nel gruppo Deployment, scegli _ HelloWorld DepGroup
18. In Tipo di revisione, scegli La mia applicazione è archiviata in Amazon S3, quindi in Posizione di revisione, inserisci la posizione della revisione dell'applicazione Hello World di esempio che hai precedentemente caricato su Amazon S3. Per ottenere la posizione:

a. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
b. Nell'elenco dei bucket, scegli amzn-s3-demo-bucket (o il nome del bucket in cui hai caricato la revisione dell'applicazione).
c. Nell'elenco degli HelloWorldoggetti App.zip, scegliete _.
d. Nella scheda Panoramica, scegliere Copia percorso.
e. Tornate alla CodeDeploy console e, in Posizione di revisione, incollate il valore del campo Link.

19. In Tipo file di revisione, selezionare .zip.
20. (Facoltativo) Inserire un commento in Deployment description (Descrizione distribuzione).
21. Scegli Create deployment (Crea distribuzione). Nella pagina Deployments (Distribuzioni) vengono visualizzate le informazioni relative alla nuova distribuzione creata.

Monitora e risolvi i problemi della distribuzione

Usa la console AWS CLI o la console per monitorare e risolvere i problemi della distribuzione.

Argomenti

- Per monitorare e risolvere problemi relativi alla distribuzione (CLI)
- Per monitorare e risolvere problemi relativi alla distribuzione (console)

Per monitorare e risolvere problemi relativi alla distribuzione (CLI)

1. Ottenere l'ID della distribuzione chiamando il comando list-deployments rispetto all'applicazione denominata **`HelloWorld_App`** e il gruppo di distribuzione denominato **`HelloWorld_DepGroup`**:

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

2. Chiamare il comando get-deployment con l'ID di distribuzione:

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

3. Il comando restituisce lo stato complessivo della distribuzione. In caso di esito positivo, il valore è `Succeeded`.

Se lo stato generale è`Failed`, puoi chiamare comandi come list-deployment-instances e get-deployment-instance per la risoluzione dei problemi. Per ulteriori opzioni di risoluzione dei problemi, consulta Analisi dei file di log per individuare gli errori di distribuzione nelle istanze.

Per monitorare e risolvere problemi relativi alla distribuzione (console)

Nella pagina Distribuzioni della console, puoi monitorare lo stato della distribuzione nella colonna Stato. CodeDeploy

Per ottenere ulteriori informazioni sulla distribuzione, in particolare se il valore della colonna Status (Stato) è diverso da Succeeded (Riuscito):

1. Nella tabella Deployments (Distribuzioni), scegliere il proprio ID di distribuzione. In caso di esito negativo di una distribuzione, nella pagina dei dettagli delle distribuzione viene visualizzato un messaggio che descrive la causa dell'errore.
2. . Vengono visualizzate ulteriori informazioni sulle istanze della distribuzione. Dopo un errore di distribuzione, potresti essere in grado di determinare su quali istanze Amazon EC2 e in quale fase l'implementazione non è riuscita.
3. Se si desidera migliorare la procedura di risoluzione dei problemi, è possibile utilizzare una tecnica come View Instance Details. Puoi anche analizzare i file di log di distribuzione su un'istanza Amazon EC2. Per ulteriori informazioni, consulta Analisi dei file di log per individuare gli errori di distribuzione nelle istanze.

Verifica la tua implementazione

Se la distribuzione ha esito positivo, verifica il funzionamento dell'installazione. Utilizza l'indirizzo DNS pubblico dell'istanza Amazon EC2 per visualizzare la pagina Web in un browser Web. (Per ottenere il valore DNS pubblico, nella console Amazon EC2, scegli l'istanza Amazon EC2 e, nella scheda Descrizione, cerca il valore in Public DNS.)

Ad esempio, se l'indirizzo DNS pubblico della tua istanza Amazon EC2 **`ec2-01-234-567-890.compute-1.amazonaws.com`** è, utilizzerai il seguente URL:

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

In caso di esito positivo, viene visualizzata una pagina Web di Hello, World!.

Passaggio 5: aggiorna e ridistribuisci il tuo «ciao, mondo!» applicazione

Ora che hai distribuito con successo la revisione dell'applicazione, sulla macchina di sviluppo, aggiorna il codice della pagina web e poi usalo CodeDeploy per ridistribuire il sito. Dopo la ridistribuzione, dovresti essere in grado di vedere le modifiche sull'istanza Amazon EC2.

Argomenti

- Modifica la pagina web
- Ridistribuire il sito

Modifica la pagina web

1. Passa alla sottocartella `c:\temp\HelloWorldApp` e utilizza un editor di testo per modificare il file `index.html`:

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

2. Rivedi il contenuto del file `index.html` per modificare il colore di sfondo e parte del testo sulla pagina Web, quindi salva il file:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/
TR/html4/loose.dtd">
<html>
<head>
  <title>Hello Again, World!</title>
  <style>
    body {
      color: #ffffff;
      background-color: #66cc00;
      font-family: Arial, sans-serif;
      font-size:14px;
    }
  </style>
</head>
<body>
  <div align="center"><h1>Hello Again, World!</h1></div>
  <div align="center"><h2>You have successfully deployed a revision of an
 application using CodeDeploy</h2></div>
  <div align="center">
```

```
    <p>What to do next? Take a look through the <a href="https://aws.amazon.com/
codedeploy">CodeDeploy Documentation</a>.</p>
  </div>
</body>
</html>
```

Ridistribuire il sito

Ora che hai modificato il codice, usa Amazon S3 e ridistribuisci CodeDeploy la pagina web.

Raggruppa e carica le modifiche su Amazon S3 come descritto in. Raggruppa i file dell'applicazione in un unico file di archivio e invia il file di archivio (Mentre segui queste istruzioni, non è necessario creare una nuova applicazione.) Assegna alla revisione la stessa chiave come in precedenza (**`HelloWorld_App.zip`**). Caricalo nello stesso bucket Amazon S3 che hai creato in precedenza (ad esempio,). **`amzn-s3-demo-bucket`**

Usa la console AWS CLI o la CodeDeploy console per ridistribuire il sito.

Argomenti

- Per ridistribuire il sito (CLI)
- Per ridistribuire il sito (console)

Per ridistribuire il sito (CLI)

Chiama il comando create-deployment per creare una distribuzione basata sulla revisione caricata, di nuovo utilizzando l'applicazione denominata **`HelloWorld_App`**, la configurazione della distribuzione denominata **`CodeDeployDefault.OneAtATime`**, il gruppo di distribuzione denominato **`HelloWorld_DepGroup`** e la revisione denominata **`HelloWorld_App.zip`** nel bucket denominato **`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
```

Puoi controllare lo stato della nuova distribuzione, come descritto in Monitora e risolvi i problemi della distribuzione.

Una CodeDeploy volta ridistribuito il sito, visita nuovamente il sito nel browser Web per verificare che il colore di sfondo e il testo della pagina Web siano stati modificati. (Potrebbe essere necessario

aggiornare il browser.) Se il colore di sfondo e il testo sono stati modificati, complimenti. Hai modificato e ridistribuito il sito.

Per ridistribuire il sito (console)

1. Accedi a Console di gestione AWS e apri la console all'indirizzo. CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione , scegliere Applications (Applicazioni).
3. Nell'elenco Applicazioni, scegli HelloWorld_App.
4. Nella scheda Deployments (Distribuzioni), scegliere Create deployment (Crea distribuzione).
   a. Nell'elenco dei gruppi di distribuzione, scegli HelloWorld_ DepGroup.
   b. In Luogo di revisione, inserisci il link Amazon S3 per la revisione.

      Per individuare il valore del link:

      i. Accedi a Console di gestione AWS e apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/

         Cerca e apri amzn-s3-demo-bucket, quindi scegli la tua revisione nella console Amazon S3. **`HelloWorld_App.zip`**

      ii. Se il riquadro Proprietà non è visibile nella console Amazon S3, scegli il pulsante Proprietà.

      iii. Nel riquadro Properties (Proprietà), copiare il valore del campo Link (Collegamento).

      iv. Torna alla CodeDeploy console, quindi incolla il link nella posizione di revisione.

      v.

   c. In Revision file type (Tipo di file della revisione), se viene visualizzato un messaggio che afferma che non è possibile rilevare il tipo di file, scegliere .zip.
   d. Lasciare vuoto il campo Deployment description (Descrizione distribuzione).
   e. Espandere Deployment group overrides (Sovrascritture gruppo di distribuzione). Nell'elenco Deployment configuration (Configurazione distribuzione), scegliere

CodeDeployDefault.OneAtATime, quindi selezionare Create deployment (Crea distribuzione).

È possibile controllare lo stato della distribuzione come descritto in Monitora e risolvi i problemi della distribuzione.

Una CodeDeploy volta ridistribuito il sito, visita nuovamente il sito nel browser Web per verificare che il colore di sfondo e il testo della pagina Web siano stati modificati. (Potrebbe essere necessario aggiornare il browser.) Se il colore di sfondo e il testo sono stati modificati, complimenti. Hai modificato e ridistribuito il sito.

Passaggio 6: ripulisci il tuo «ciao mondo!» applicazione e risorse correlate

Ora hai effettuato con successo un aggiornamento a «Hello, World!» codificare e ridistribuire il sito. Per evitare addebiti continuativi per risorse create per completare questo tutorial, devi eliminare:

- Qualsiasi CloudFormation stack (o termina qualsiasi istanza Amazon EC2, se le hai create all'esterno). CloudFormation
- Qualsiasi bucket Amazon S3.
- L'applicazione `HelloWorld_App` in CodeDeploy.
- L'associazione AWS Systems Manager State Manager dell' CodeDeploy agente.

Puoi utilizzare Amazon S3 CloudFormation, Amazon EC2 CodeDeploy e console o le API per eseguire AWS la pulizia. AWS CLI

Argomenti

- Come procedere all'eliminazione delle risorse (CLI)
- Per cancellare le risorse (console)
- Fasi successive

Come procedere all'eliminazione delle risorse (CLI)

1. Se hai usato lo CloudFormation stack per questo tutorial, elimina lo stack chiamando il comando sullo stack denominato. delete-stack **`CodeDeployDemoStack`** Ciò interrompe tutte le istanze Amazon EC2 associate ed elimina tutti i ruoli IAM associati originariamente creati dallo stack.

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

2. Per eliminare il bucket Amazon S3, chiama il rm comando con lo --recursive switch accanto al bucket denominato. **`amzn-s3-demo-bucket`** Il bucket e tutti gli oggetti in esso contenuti saranno quindi eliminati.

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

3. Per eliminare l'`HelloWorld_App`applicazione da CodeDeploy, chiama il comando. delete-application Così facendo, saranno eliminati tutti i record del gruppo di distribuzione associato e i record di distribuzione dell'applicazione.

```
aws deploy delete-application --application-name HelloWorld_App
```

4. Per eliminare l'associazione Systems Manager State Manager, chiamare il delete-association comando.

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

È possibile ottenerla *`association-id`* chiamando il describe-association comando.

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

5. Se non hai utilizzato lo CloudFormation stack per questo tutorial, chiama il terminate-instances comando per terminare le istanze Amazon EC2 che hai creato manualmente. Fornisci l'ID dell'istanza Amazon EC2 da terminare.

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

Per cancellare le risorse (console)

Se hai utilizzato il nostro CloudFormation modello per questo tutorial, elimina lo stack associato CloudFormation .

1. Accedi a Console di gestione AWS e apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.

2. Nella casella di ricerca, digita il nome CloudFormation dello stack (ad esempio,**`CodeDeployDemoStack`**).
3. Seleziona la casella posta accanto al nome dello stack.
4. Nel menu Actions (Operazioni), scegli Delete Stack (Elimina stack). Ciò elimina lo stack, termina tutte le istanze Amazon EC2 associate ed elimina tutti i ruoli IAM associati.

Per terminare le istanze Amazon EC2 che hai creato al di fuori di uno stack: CloudFormation

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Nell'area Instances (Istanze), scegliere proprio Instances (Istanze).
3. Nella casella di ricerca, digita il nome dell'istanza Amazon EC2 che desideri terminare, quindi premi Invio.
4. Scegli l'istanza Amazon EC2.
5. Selezionare Actions (Operazioni), passare a Instance State (Stato istanza), quindi selezionare Terminate (Termina). Quando richiesto, scegliere Yes, Terminate (Sì, termina). Ripeti questi passaggi per eventuali istanze Amazon EC2 aggiuntive.

Per eliminare il bucket Amazon S3:

1. Accedi a Console di gestione AWS e apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Nell'elenco dei bucket, cerca e scegli il nome del bucket Amazon S3 (ad esempio,). **`amzn-s3-demo-bucket`**
3. Per eliminare un bucket, occorre innanzitutto rimuoverne i contenuti. Selezionare tutti i file nel bucket, ad esempio **`HelloWorld_App.zip`**. Dal menu Actions (Operazioni), scegliere Delete (Elimina). Quando viene richiesto di confermare l'eliminazione, scegliere OK (OK).
4. Una volta svuotato, il bucket può essere eliminato. Nell'elenco dei bucket, selezionare la riga del bucket d'interesse (non il suo nome). Selezionare Delete bucket (Elimina bucket) e, quando viene richiesto di confermare, selezionare OK (OK).

Per eliminare l'applicazione da: `HelloWorld_App` CodeDeploy

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Scegli **`HelloWorld_App`**.
4. Scegli Elimina applicazione.
5. Quando richiesto, digitare **`Delete`**, quindi scegliere Delete (Elimina).

Per eliminare l'associazione Systems Manager State Manager:

1. Aprire la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager.
2. Nel riquadro di navigazione, seleziona State Manager.
3. Scegli l'associazione creata e seleziona Elimina.

Fasi successive

Se sei arrivato qui, hai completato con successo una distribuzione con CodeDeploy. Complimenti!

Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux)

Questo tutorial ti aiuta ad acquisire esperienza CodeDeploy guidandoti nella distribuzione di una revisione di esempio dell'applicazione su una singola istanza locale, ovvero un dispositivo fisico che non è un'istanza Amazon EC2, che esegue Windows Server, Ubuntu Server o Red Hat Enterprise Linux (RHEL). Per informazioni sulle istanze locali e su come funzionano, consulta. CodeDeploy Working with On-Premises Instances

Non hai trovato ciò che cerchi?

- Per fare pratica con la distribuzione su un'istanza Amazon EC2 che esegue Amazon Linux o RHEL, consulta. Tutorial: Esegui WordPress la distribuzione su un'istanza Amazon EC2 (Amazon Linux o Red Hat Enterprise Linux e Linux, macOS o Unix)
- Per fare pratica con la distribuzione su un'istanza Amazon EC2 che esegue Windows Server, consulta. Tutorial: distribuisci un messaggio «ciao, mondo!» applicazione con CodeDeploy (Windows Server)

Argomenti

- Prerequisiti
- Passaggio 1: configurare l'istanza locale
- Passaggio 2: creare una revisione di esempio dell'applicazione
- Fase 3: raggruppa e carica la revisione dell'applicazione su Amazon S3
- Fase 4: Implementa la revisione dell'applicazione
- Passaggio 5: verifica la distribuzione
- Fase 6: eliminare le risorse

Prerequisiti

Prima di iniziare questo tutorial, devi completare i prerequisiti di cui sopraGuida introduttiva con CodeDeploy, che includono la configurazione di un utente, l'installazione o l'aggiornamento di e la creazione di un ruolo di servizio AWS CLI. Non è necessario creare un profilo di istanza IAM come descritto nei prerequisiti. On-premises le istanze non utilizzano profili di istanza IAM.

Il dispositivo fisico da configurare come istanza locale deve eseguire uno dei sistemi operativi elencati in Sistemi operativi supportati dall'agente CodeDeploy.

Passaggio 1: configurare l'istanza locale

Per poter procedere alla distribuzione, occorre innanzitutto configurare l'istanza locale. Segui le istruzioni riportate in Working with On-Premises Instances, poi torna a questa pagina.

Installa l'agente CodeDeploy

Dopo aver configurato l'istanza locale, segui i passaggi per le istanze locali in Installare l' CodeDeploy agente e torna a questa pagina.

Passaggio 2: creare una revisione di esempio dell'applicazione

Questa fase prevede la creazione di una revisione di applicazione esemplificativa da distribuire sull'istanza locale.

Poiché è difficile sapere quali software e funzionalità sono già installati, o quali sono autorizzati dalle politiche dell'organizzazione, sull'istanza locale, la revisione di esempio dell'applicazione che offriamo qui utilizza semplicemente script batch (per Windows Server) o script di shell (per Ubuntu Server e RHEL) per scrivere file di testo in una posizione sull'istanza locale. Viene scritto un file per ciascuno dei diversi eventi del ciclo di vita della CodeDeploy distribuzione, tra cui Install AfterInstall, ApplicationStart, e ValidateService. Durante l'evento del ciclo di vita della BeforeInstalldistribuzione, verrà eseguito uno script per rimuovere i vecchi file scritti durante le distribuzioni precedenti di questo esempio e creare una posizione sull'istanza locale in cui scrivere i nuovi file.

> **Note**
>
> La distribuzione di questa revisione di applicazione esemplificativa potrebbe non riuscire in presenza di una delle condizioni seguenti:
>
> - L'utente che avvia l' CodeDeploy agente sull'istanza locale non dispone dell'autorizzazione per eseguire script.
> - L'utente non dispone dell'autorizzazione per creare o eliminare cartelle nelle posizioni elencate negli script.
> - L'utente non è autorizzato a creare file di testo nelle posizioni elencate negli script.

> **Note**
>
> Se hai configurato un'istanza di Windows Server e desideri distribuire un campione diverso, potresti voler utilizzare quello illustrato Fase 2: configura il contenuto sorgente da distribuire sull'istanza Amazon EC2 di Windows Server nel Tutorial: distribuisci un messaggio «ciao, mondo!» applicazione con CodeDeploy (Windows Server) tutorial.
> Se hai configurato un'istanza RHEL e desideri distribuire un esempio diverso, potresti voler utilizzare quello illustrato Fase 2: configura il contenuto sorgente da distribuire sull'istanza Amazon Linux o Red Hat Enterprise Linux Amazon EC2 nel tutorial. Tutorial: Esegui WordPress la distribuzione su un'istanza Amazon EC2 (Amazon Linux o Red Hat Enterprise Linux e Linux, macOS o Unix)

Attualmente non esiste un esempio alternativo per Ubuntu Server.

1. Nel computer di sviluppo, crea innanzitutto una sottodirectory (sottocartella) denominata `CodeDeployDemo-OnPrem` che archivi i file della revisione di applicazione esemplificativa, poi accedi a tale sottocartella. Per questo esempio, supponiamo che utilizzerai la `c:\temp` cartella come cartella principale per Windows Server o la `/tmp` cartella come cartella principale per Ubuntu Server e RHEL. È possibile utilizzare una cartella diversa, prestando attenzione a sostituirla alla nostra coerentemente nel corso dell'intero tutorial:

   Per Windows:

   ```
   mkdir c:\temp\CodeDeployDemo-OnPrem
   cd c:\temp\CodeDeployDemo-OnPrem
   ```

   Per Linux, macOS o Unix:

   ```
   mkdir /tmp/CodeDeployDemo-OnPrem
   cd /tmp/CodeDeployDemo-OnPrem
   ```

2. Nella radice della sottocartella `CodeDeployDemo-OnPrem`, utilizza un editor di testo per creare due file denominati `appspec.yml` e `install.txt`:

   `appspec.yml`per 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`per Ubuntu Server e RHEL:

```
version: 0.0
os: linux
files:
  - source: ./install.txt
    destination: /tmp/CodeDeployExample
hooks:
  BeforeInstall:
    - location: ./scripts/before-install.sh
      timeout: 900
  AfterInstall:
    - location: ./scripts/after-install.sh
      timeout: 900
  ApplicationStart:
    - location: ./scripts/application-start.sh
      timeout: 900
  ValidateService:
    - location: ./scripts/validate-service.sh
      timeout: 900
```

Per ulteriori informazioni sui AppSpec file, vedere Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy eCodeDeploy AppSpec riferimento al file.

`install.txt`:

```
The Install deployment lifecycle event successfully completed.
```

3. Nella radice della sottocartella `CodeDeployDemo-OnPrem`, crea una sottocartella `scripts` per poi accedervi:

   Per Windows:

   ```
   mkdir c:\temp\CodeDeployDemo-OnPrem\scripts
   cd c:\temp\CodeDeployDemo-OnPrem\scripts
   ```

   Per Linux, macOS o Unix:

```
mkdir -p /tmp/CodeDeployDemo-OnPrem/scripts
cd /tmp/CodeDeployDemo-OnPrem/scripts
```

4. Nella radice della `scripts` sottocartella, utilizzate un editor di testo per creare quattro file denominati `before-install.batafter-install.bat`,`application-start.bat`,, e `validate-service.bat` per Windows Server, `before-install.sh` `after-install.shapplication-start.sh`, e `validate-service.sh` per Ubuntu Server e RHEL:

   Per 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
```

Per Ubuntu Server e RHEL:

`before-install.sh`:

```
#!/bin/bash
export FOLDER=/tmp/CodeDeployExample

if [ -d $FOLDER ]
then
 rm -rf $FOLDER
fi

mkdir -p $FOLDER
```

`after-install.sh`:

```
#!/bin/bash
cd /tmp/CodeDeployExample

echo "The AfterInstall deployment lifecycle event successfully completed." > after-
install.txt
```

`application-start.sh`:

```
#!/bin/bash
cd /tmp/CodeDeployExample

echo "The ApplicationStart deployment lifecycle event successfully completed." >
 application-start.txt
```

`validate-service.sh`:

```
#!/bin/bash
cd /tmp/CodeDeployExample

echo "The ValidateService deployment lifecycle event successfully completed." >
 validate-service.txt
```

```
unset FOLDER
```

5. Solo per Ubuntu Server e RHEL, assicurati che i quattro script di shell abbiano i permessi di esecuzione:

```
chmod +x ./scripts/*
```

Fase 3: raggruppa e carica la revisione dell'applicazione su Amazon S3

Prima di poter distribuire la revisione dell'applicazione, dovrai raggruppare i file e quindi caricare il pacchetto di file in un bucket Amazon S3. A tale scopo, segui le istruzioni riportate in Crea un'applicazione con CodeDeploy e Invia una revisione CodeDeploy ad Amazon S3 EC2/On-Premises (solo distribuzioni). (Consigliamo di utilizzare i nomi `CodeDeploy-OnPrem-App` per l'applicazione e `CodeDeploy-OnPrem-DG` per il gruppo di distribuzione, nonostante la denominazione sia libera, a discrezione dell'utente.) Dopo avere effettuato quanto riportato nelle istruzioni, torna a questa pagina.

> **Note**
>
> In alternativa, puoi caricare il pacchetto di file in un GitHub repository e distribuirlo da lì. Per ulteriori informazioni, consulta Integrazione con CodeDeploy GitHub.

Fase 4: Implementa la revisione dell'applicazione

Dopo aver caricato la revisione dell'applicazione in un bucket Amazon S3, prova a distribuirla sulla tua istanza locale. Segui le istruzioni riportate in Crea una distribuzione con CodeDeploy, poi torna a questa pagina.

Passaggio 5: verifica la distribuzione

Per verificare l'avvenuta distribuzione, segui le istruzioni riportate in Visualizza i dettagli CodeDeploy della distribuzione, poi torna a questa pagina.

Se la distribuzione ha avuto successo, troverai quattro file di testo nella `c:\temp\CodeDeployExample` cartella (per Windows Server) o `/tmp/CodeDeployExample` (per Ubuntu Server e RHEL).

Se la distribuzione non riesce, segui le istruzioni per la risoluzione dei problemi riportate in View Instance Details e Risoluzione dei problemi relativi alle istanze. Apporta quindi le necessarie

modifiche correttive, aggrega e carica nuovamente la revisione di applicazione e, infine, ritenta la distribuzione.

Fase 6: eliminare le risorse

Per evitare addebiti continui per le risorse che hai creato per questo tutorial, elimina il bucket Amazon S3 se non lo utilizzerai più. Puoi anche pulire le risorse associate, come i record dell'applicazione e del gruppo di distribuzione CodeDeploy e l'istanza locale.

Puoi utilizzare AWS CLI o una combinazione delle console Amazon S3 CodeDeploy e AWS CLI per ripulire le risorse.

Risorse di pulizia (CLI)

Per eliminare il bucket Amazon S3

- Chiamare il comando rm e l'opzione `--recursive` per il bucket (ad esempio, `amzn-s3-demo-bucket`). Il bucket e tutti gli oggetti in esso contenuti saranno eliminati.

```
aws s3 rm s3://your-bucket-name --recursive --region region
```

Per eliminare i record dell'applicazione e del gruppo di distribuzione in CodeDeploy

- Richiamate il comando delete-application sull'applicazione (ad esempio,`CodeDeploy-OnPrem-App`). I record della distribuzione e del gruppo di distribuzione saranno eliminati.

```
aws deploy delete-application --application-name your-application-name
```

Per annullare la registrazione dell'istanza locale ed eliminare l'utente IAM

- Chiama il comando deregister sull'istanza e sulla regione locali:

```
aws deploy deregister --instance-name your-instance-name --delete-iam-user --
region your-region
```

> **Note**
>
> Se non desideri eliminare l'utente IAM associato a questa istanza locale, utilizza invece l'opzione. `--no-delete-iam-user`

Per disinstallare l' CodeDeploy agente e rimuovere il file di configurazione dall'istanza locale

- Dall'istanza locale, chiamate il comando uninstall:

```
aws deploy uninstall
```

A questo punto, la procedura di eliminazione delle risorse impiegate per il tutorial è completa e conclusa.

Pulisci le risorse (console)

Per eliminare il bucket Amazon S3

1. Accedi Console di gestione AWS e apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Selezionare l'icona posta accanto al bucket da eliminare (ad esempio, `amzn-s3-demo-bucket`), ma non selezionare il bucket stesso.
3. Scegli Azioni, quindi Elimina.
4. Quando viene richiesto di eliminare il bucket, scegliere OK (OK).

Per eliminare i record dell'applicazione e del gruppo di distribuzione in CodeDeploy

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, scegliere Applications (Applicazioni).

3. Scegliere il nome dell'applicazione che si desidera eliminare (ad esempio, `CodeDeploy-OnPrem-App`), quindi selezionare Delete (Elimina).
4. Quando richiesto, digitare il nome dell'applicazione per confermare che si desidera eliminarla, quindi scegliere Delete (Elimina).

Non è possibile utilizzare la AWS CodeDeploy console per annullare la registrazione dell'istanza locale o disinstallare l'agente. CodeDeploy Segui le istruzioni in Per annullare la registrazione dell'istanza locale ed eliminare l'utente IAM.

Tutorial: CodeDeploy Da utilizzare per distribuire un'applicazione in un gruppo di Auto Scaling

In questo tutorial, utilizzerai CodeDeploy per distribuire una revisione dell'applicazione in un gruppo Auto Scaling. Amazon EC2 Auto Scaling avvia le istanze Amazon EC2 utilizzando condizioni predefinite, quindi termina tali istanze quando non sono più necessarie. Amazon EC2 Auto Scaling CodeDeploy può contribuire alla scalabilità garantendo che disponga sempre del numero corretto di istanze Amazon EC2 disponibili per gestire il carico per le distribuzioni. Per informazioni sull' CodeDeployintegrazione di Amazon EC2 Auto Scaling con, consulta. Integrazione CodeDeploy con Amazon EC2 Auto Scaling

Argomenti

- Prerequisiti
- Fase 1: Creare e configurare il gruppo Auto Scaling
- Fase 2: Distribuire l'applicazione nel gruppo Auto Scaling
- Passaggio 3: verifica i risultati
- Fase 4: Aumentare il numero di istanze Amazon EC2 nel gruppo Auto Scaling
- Fase 5: Controlla nuovamente i risultati
- Fase 6: pulizia

Prerequisiti

Per seguire questo tutorial:

- Completa tutti i passaggiGuida introduttiva con CodeDeploy, incluse l'impostazione e la configurazione e la AWS CLI creazione di un profilo di istanza IAM (**`CodeDeployDemo-EC2-`**

`Instance-Profile`) e di un ruolo di servizio (). **`CodeDeployDemo`** Un ruolo di servizio è un tipo speciale di ruolo IAM che concede l'autorizzazione di servizio ad agire per tuo conto.

- Se crei il tuo gruppo Auto Scaling con un modello di avvio, devi aggiungere le seguenti autorizzazioni:
  - `ec2:RunInstances`
  - `ec2:CreateTags`
  - `iam:PassRole`

  Per ulteriori informazioniFase 2: Creare un ruolo di servizio, consulta Creazione di un modello di lancio per un gruppo Auto Scaling e Supporto dei modelli di avvio nella Guida per l'utente di Amazon EC2 Auto Scaling.
- Crea e usa una revisione compatibile con un'istanza di Ubuntu Server e. CodeDeploy Per la tua revisione, puoi eseguire una delle seguenti operazioni:
  - Creare e utilizzare la revisione di esempio in Passaggio 2: creare una revisione di esempio dell'applicazione nel tutorial Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux).
  - Per creare una tua revisione, consultaUtilizzo delle revisioni delle applicazioni per CodeDeploy.
- Crea un gruppo di sicurezza denominato **`CodeDeployDemo-AS-SG`** con la seguente regola in entrata:
  - Tipo: HTTP
  - Fonte: Anywhere

  Ciò è necessario per visualizzare l'applicazione e verificare il successo dell'implementazione. Per informazioni su come creare un gruppo di sicurezza, consulta Creazione di un gruppo di sicurezza nella guida per l'utente di Amazon EC2.

Fase 1: Creare e configurare il gruppo Auto Scaling

In questo passaggio, creerai un gruppo di Auto Scaling che contiene una singola istanza Amazon Amazon EC2 di Amazon Linux, RHEL o Windows Server. In una fase successiva, indicherai ad Amazon EC2 Auto Scaling di aggiungere un'altra istanza Amazon EC2 CodeDeploy e di distribuirvi la tua revisione.

Argomenti

- Per creare e configurare il gruppo Auto Scaling (CLI)
- Per creare e configurare il gruppo Auto Scaling (console)

Per creare e configurare il gruppo Auto Scaling (CLI)

1. Chiama il create-launch-template comando per creare un modello di lancio di Amazon EC2.

   Prima di chiamare questo comando, è necessario l'ID di un AMI che funzioni per questo tutorial, rappresentato dal segnaposto*image-id*. È inoltre necessario il nome di una coppia di chiavi di istanza Amazon EC2 per abilitare l'accesso all'istanza Amazon EC2, rappresentata dal segnaposto. *key-name*

   Per ottenere l'ID di un'AMI da usare con questo tutorial:

   a. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.

   b. Nel pannello di navigazione, in Istanze, scegli Istanze, quindi scegli Launch Instance.

   c. Nella scheda Quick Start della pagina Scegli un'immagine di Amazon Machine, annota l'ID dell'AMI accanto all'AMI Amazon Linux 2, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS o Microsoft Windows Server 2012 R2.

   > **Note**
   >
   > Se si dispone di una versione personalizzata di un'AMI compatibile con CodeDeploy, sceglierla qui invece di cercarla usando la scheda Quick Start (Avvio rapido). Per informazioni sull'utilizzo di un'AMI personalizzata con CodeDeploy Amazon EC2 Auto Utilizzo di un'AMI personalizzata con CodeDeploy Amazon EC2 Auto Scaling Scaling, consulta.

   Per la coppia di chiavi dell'istanza Amazon EC2, usa il nome della coppia di chiavi dell'istanza Amazon EC2.

   Chiamare il comando create-launch-template.

   Su macchine Linux, macOS o Unix locali:

```
aws ec2 create-launch-template \
  --launch-template-name CodeDeployDemo-AS-Launch-Template \
```

```
  --launch-template-data file://config.json
```

Il contenuto del file: `config.json`

```
{
  "InstanceType":"t1.micro",
  "ImageId":"image-id",
  "IamInstanceProfile":{
    "Name":"CodeDeployDemo-EC2-Instance-Profile"
  },
  "KeyName":"key-name"
}
```

Nei computer Windows locali:

```
aws ec2 create-launch-template --launch-template-name CodeDeployDemo-AS-Launch-
Template --launch-template-data file://config.json
```

Il contenuto del `config.json` file:

```
{
  "InstanceType":"t1.micro",
  "ImageId":"image-id",
  "IamInstanceProfile":{
    "Name":"CodeDeployDemo-EC2-Instance-Profile"
  },
  "KeyName":"key-name"
}
```

Questi comandi, insieme al `config.json` file, creano un modello di avvio di Amazon EC2 denominato CodeDeployDemo-AS-Launch-Template per il gruppo Auto Scaling che verrà creato in un passaggio successivo in base al tipo di istanza Amazon EC2 t1.micro. In base all'input fornito per e `ImageId` `IamInstanceProfile``KeyName`, il modello di lancio specifica anche l'ID AMI, il nome del profilo dell'istanza associato al ruolo IAM da passare alle istanze al momento del lancio e la coppia di chiavi Amazon EC2 da utilizzare per la connessione alle istanze.

2. Chiama il create-auto-scaling-group comando per creare un gruppo Auto Scaling. Avrai bisogno del nome di una delle zone di disponibilità in una delle regioni elencate in Regione e degli endpoint in Riferimenti generali di AWS, rappresentati dal segnaposto. *`availability-zone`*

> **Note**
>
> Per visualizzare un elenco delle zone di disponibilità in una regione, chiama:
>
> ```
> aws ec2 describe-availability-zones --region region-name
> ```
>
> Ad esempio, per visualizzare un elenco di zone di disponibilità nella regione Stati Uniti occidentali (Oregon), chiama:
>
> ```
> aws ec2 describe-availability-zones --region us-west-2
> ```
>
> Per un elenco degli identificatori dei nomi delle regioni, vedere. Nomi dei bucket del Resource Kit per regione

Su macchine Linux, macOS o Unix locali:

```
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
```

Nei computer Windows locali:

```
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
```

Questi comandi creano un gruppo Auto Scaling denominato in **`CodeDeployDemo-AS-Group`** base al modello di lancio di Amazon EC2 denominato. **`CodeDeployDemo-AS-Launch-Template`** Questo gruppo di Auto Scaling ha una sola istanza Amazon EC2 e viene creato nella zona di disponibilità specificata. Ogni istanza in questo gruppo Auto Scaling avrà il tag.

`Name=CodeDeployDemo` Il tag verrà utilizzato quando si installerà l' CodeDeploy agente in un secondo momento.

3. Chiamare il comando describe-auto-scaling-groups per **`CodeDeployDemo-AS-Group`**:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
 CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,
 LifecycleState]" --output text
```

Non procedere finché i valori restituiti non mostrano `Healthy` e `InService`.

4. L'agente deve essere installato sulle istanze del gruppo Auto Scaling per poter essere utilizzate nelle CodeDeploy distribuzioni. CodeDeploy Installa l' CodeDeploy agente chiamando il create-association comando from AWS Systems Manager con i tag aggiunti quando è stato creato il gruppo Auto Scaling.

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

 --parameters action=Install, name=AWSCodeDeployAgent \
  --schedule-expression "cron(0 2 ? * SUN *)"
```

Questo comando crea un'associazione in Systems Manager State Manager che installerà l' CodeDeploy agente su tutte le istanze del gruppo Auto Scaling e quindi tenterà di aggiornarlo alle 2:00 ogni domenica mattina. Per ulteriori informazioni sull' CodeDeploy agente, vedere Lavorare con l'agente. CodeDeploy Per ulteriori informazioni su Systems Manager, vedere What is AWS Systems Manager.

Per creare e configurare il gruppo Auto Scaling (console)

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nella barra di navigazione globale, assicurati che sia selezionata una delle regioni elencate in Regione ed endpoint in. Riferimenti generali di AWS Le risorse di Amazon EC2 Auto Scaling sono legate alla regione specificata CodeDeploy e sono supportate solo in alcune regioni.
3. Nella barra di navigazione, in Istanze, scegli Launch Templates.
4. Scegli Crea modello di avvio.

5. Nella finestra di dialogo Avvia nome e descrizione del modello, per Avvia nome modello, inserisci**`CodeDeployDemo-AS-Launch-Template`**. Lascia i valori predefiniti per gli altri campi.
6. Nella finestra di dialogo Amazon machine image (AMI), fai clic sul menu a discesa sotto AMI, scegli un AMI che funzioni con questo tutorial:

   - Nella scheda Quick Start del menu a discesa AMI, scegli una delle seguenti opzioni: Amazon Linux 2 AMI, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS o Microsoft Windows Server 2012 R2.

   > **Note**
   >
   > Se si dispone di una versione personalizzata di un'AMI compatibile con CodeDeploy, sceglierla qui invece di cercarla usando la scheda Quick Start (Avvio rapido). Per informazioni sull'utilizzo di un'AMI personalizzata con CodeDeploy Amazon EC2 Auto Utilizzo di un'AMI personalizzata con CodeDeploy Amazon EC2 Auto Scaling Scaling, consulta.

7. In Tipo di istanza, seleziona il menu a discesa e scegli t1.micro. Puoi utilizzare la barra di ricerca per trovarlo più rapidamente.
8. Nella finestra di dialogo Key pair (login), seleziona Scegli una coppia di chiavi esistente. Nell'elenco a discesa Seleziona una coppia di chiavi, scegli la coppia di chiavi di istanza Amazon EC2 che hai creato o utilizzato nei passaggi precedenti.
9. Nella finestra di dialogo Impostazioni di rete, scegli Virtual Public Cloud (VPC).

   Nel menu a discesa Gruppi di sicurezza, scegli il gruppo di sicurezza che hai creato nella sezione dei prerequisiti del tutorial (). **`CodeDeployDemo-AS-SG`**
10. Espandi la finestra di dialogo Dettagli avanzati. Nel menu a discesa del profilo dell'istanza IAM, seleziona il ruolo IAM che hai creato in precedenza (**`CodeDeployDemo-EC2-Instance-Profile`**) nel profilo dell'istanza IAM.

    Lascia il resto delle impostazioni predefinite.
11. Scegli Crea modello di avvio.
12. Nella finestra di dialogo Passaggi successivi, scegliete Crea gruppo Auto Scaling.
13. Nella pagina Scegli il modello di avvio o la configurazione, per il nome del gruppo Auto Scaling, digitare. **`CodeDeployDemo-AS-Group`**

14. Nella finestra di dialogo Avvia modello, il modello di avvio (**`CodeDeployDemo-AS-Launch-Template`**) deve essere compilato; in caso contrario, selezionalo dal menu a discesa. Lascia le impostazioni predefinite e scegli Avanti.
15. Nella pagina Scegli le opzioni di avvio dell'istanza, nella sezione Rete, per VPC, scegli il VPC predefinito. Quindi, per le zone di disponibilità e le sottoreti, scegli una sottorete predefinita. È necessario creare un VPC se non è possibile scegliere l'impostazione predefinita. Per ulteriori informazioni, consulta la sezione Guida introduttiva ad Amazon VPC.
16. Nella sezione Instance type requirements (Requisiti del tipo di istanza), utilizza l'impostazione di default per semplificare questo passaggio. (Non sovrascrivere il modello di avvio.) In questo tutorial, lancerai solo On-Demand le istanze utilizzando il tipo di istanza specificato nel modello di lancio.
17. Scegli Next (Avanti) per accedere alla pagina Configure advanced options (Configura opzioni avanzate).
18. Mantieni i valori predefiniti e scegli Avanti.
19. Nella pagina Configura la dimensione del gruppo e le politiche di ridimensionamento, mantieni i valori predefiniti di dimensione del gruppo pari a 1. Scegli Next (Successivo).
20. Salta il passaggio per la configurazione delle notifiche e scegli Avanti.
21. Nella pagina Aggiungi tag, aggiungi un tag da utilizzare per l'installazione successiva dell' CodeDeploy agente. Seleziona Aggiungi tag.

    a. In Key (Chiave), immettere **`Name`**.

    b. In Valore, immetti **`CodeDeployDemo`**.

    Scegli Next (Successivo).
22. Controlla le informazioni sul gruppo Auto Scaling nella pagina Revisione, quindi scegli Crea gruppo Auto Scaling.
23. Nella barra di navigazione, con Auto Scaling Groups selezionato, sceglietе**`CodeDeployDemo-AS-Group`**, quindi scegliete la scheda Gestione istanze. Non procedere finché non InServiceviene visualizzato il valore di nella colonna Lifecycle e il valore di Healthy nella colonna Health Status.
24. Installa l' CodeDeploy agente seguendo la procedura descritta in Installare l' CodeDeploy agente e utilizzando i tag di `Name=CodeDeployDemo` istanza.

Fase 2: Distribuire l'applicazione nel gruppo Auto Scaling

In questo passaggio, distribuirai la revisione sulla singola istanza Amazon EC2 nel gruppo Auto Scaling.

Argomenti

- Per creare la distribuzione (CLI)
- Per creare la distribuzione (console)

Per creare la distribuzione (CLI)

1. Chiamare il comando create-application per creare un'applicazione denominata **`SimpleDemoApp`**:

   ```
   aws deploy create-application --application-name SimpleDemoApp
   ```

2. È necessario avere già creato un ruolo del servizio seguendo le istruzioni in Fase 2: Creare un ruolo di servizio per CodeDeploy. Il ruolo di servizio ti consentirà CodeDeploy di accedere alle tue istanze Amazon EC2 per espandere (leggere) i relativi tag. È necessario l'ARN del ruolo del servizio. Per ottenere l'ARN del ruolo del servizio, seguire le istruzioni contenute in Ottieni il ruolo di servizio ARN (CLI).
3. Ora che avete un ruolo di servizio ARN, chiamate il create-deployment-group comando per creare un gruppo di distribuzione denominato**`SimpleDemoDG`**, associato all'applicazione denominata**`SimpleDemoApp`**, utilizzando il gruppo Auto Scaling denominato e la configurazione **`CodeDeployDefault.OneAtATime`** di distribuzione **`CodeDeployDemo-AS-Group`** denominata, con il ruolo di servizio specificato ARN.

   > **Note**
   >
   > Il comando create-deployment-group fornisce supporto per la creazione di trigger che determinano l'invio di notifiche Amazon SNS agli abbonati dell'argomento su eventi specifici nelle distribuzioni e nelle istanze. Il comando supporta anche opzioni per il rollback automatico delle distribuzioni e l'impostazione di allarmi per interrompere le distribuzioni quando vengono raggiunte le soglie di monitoraggio negli allarmi Amazon. CloudWatch I comandi per queste azioni non sono inclusi in questo tutorial.

Su macchine Linux, macOS o Unix locali:

```
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
```

Nei computer Windows locali:

```
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. Chiamare il comando create-deployment per creare una distribuzione associata all'applicazione denominata **SimpleDemoApp**, la configurazione della distribuzione denominata **CodeDeployDefault.OneAtATime**, il gruppo di distribuzione denominato **SimpleDemoDG** utilizzando la revisione nel percorso specificato.

   Per Amazon Linux e RHEL, istanze Amazon EC2, chiamate da macchine Linux, macOS o Unix locali

   ```
   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*è il nome del bucket Amazon S3 che contiene i file CodeDeploy Resource Kit per la tua regione. Ad esempio, per la regione Stati Uniti orientali (Ohio), sostituisci con. *bucket-name* `aws-codedeploy-us-east-2` Per un elenco dei nomi dei bucket, vedi. Nomi dei bucket del Resource Kit per regione

   Per Amazon Linux e RHEL, istanze Amazon EC2, chiamate da macchine Windows locali

```
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*è il nome del bucket Amazon S3 che contiene i file CodeDeploy Resource Kit per la tua regione. Ad esempio, per la regione Stati Uniti orientali (Ohio), sostituisci con. *bucket-name* `aws-codedeploy-us-east-2` Per un elenco dei nomi dei bucket, vedi. Nomi dei bucket del Resource Kit per regione

Per istanze Windows Server Amazon EC2, chiamate da macchine Linux, macOS o Unix locali

```
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*è il nome del bucket Amazon S3 che contiene i file CodeDeploy Resource Kit per la tua regione. Ad esempio, per la regione Stati Uniti orientali (Ohio), sostituisci con. *bucket-name* `aws-codedeploy-us-east-2` Per un elenco dei nomi dei bucket, vedi. Nomi dei bucket del Resource Kit per regione

Per istanze Windows Server Amazon EC2, chiamate da macchine Windows locali

```
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*è il nome del bucket Amazon S3 che contiene i file CodeDeploy Resource Kit per la tua regione. Ad esempio, per la regione Stati Uniti orientali (Ohio), sostituisci con. *bucket-name* `aws-codedeploy-us-east-2` Per un elenco dei nomi dei bucket, vedi. Nomi dei bucket del Resource Kit per regione

> **Note**
>
> Attualmente, CodeDeploy non fornisce una revisione di esempio da distribuire su istanze Amazon EC2 di Ubuntu Server. Per creare una tua revisione, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.

5. Chiamare il comando get-deployment per verificare che la distribuzione sia stata completata.

   Prima di chiamare questo comando, è necessario l'ID della distribuzione che dovrebbe essere stato restituito dalla chiamata al comando create-deployment. Per ottenere nuovamente l'ID della distribuzione, chiamare il comando list-deployments per l'applicazione **`SimpleDemoApp`** e il gruppo di distribuzione con nome **`SimpleDemoDG`**:

   ```
   aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-
   name SimpleDemoDG --query "deployments" --output text
   ```

   Ora chiamare il comando get-deployment usando l'ID della distribuzione:

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

   Non continuare finché il valore restituito non è `Succeeded`.

Per creare la distribuzione (console)

1. È necessario avere già creato un ruolo del servizio seguendo le istruzioni in Fase 2: Creare un ruolo di servizio per CodeDeploy. Il ruolo di servizio consentirà di accedere CodeDeploy alle istanze per espandere (leggere) i relativi tag. Prima di utilizzare la CodeDeploy console per distribuire la revisione dell'applicazione, è necessario il ruolo di servizio ARN. Per ottenere l'ARN del ruolo del servizio, seguire le istruzioni contenute in Ottieni il ruolo di servizio ARN (console).
2. Ora che hai il ruolo di servizio ARN, puoi utilizzare la CodeDeploy console per distribuire la revisione dell'applicazione.

   Accedi a Console di gestione AWS e apri la console all' CodeDeploy indirizzo. https://console.aws.amazon.com/codedeploy

> **Note**
>
> Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

3. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
4. Scegli Crea applicazione.
5. Selezionare Custom application (Applicazione personalizzata).
6. In Application name (Nome applicazione), immettere **`SimpleDemoApp`**.
7. Nella piattaforma Compute, scegli. EC2/On-premises
8. Scegli Crea applicazione.
9. Nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione).
10. In Deployment group name (Nome del gruppo di distribuzione), immettere **`SimpleDemoDG`**.
11. In Service Role (Ruolo del servizio), scegliere il nome del ruolo del servizio.
12. In Tipo di distribuzione, scegli In-place.
13. In Configurazione ambiente seleziona Gruppi Auto Scaling, quindi scegli. **`CodeDeployDemo-AS-Group`**
14. In Deployment configuration (Configurazione della distribuzione), scegliere CodeDeployDefault.OneAtATime.
15. Deselezionare Enable load balancing (Abilita bilanciamento del carico).
16. Scegliere Create deployment group (Crea gruppo di distribuzione).
17. Nella pagina del gruppo di distribuzione, scegliere Create deployment (Crea distribuzione).
18. In Tipo di revisione, scegli La mia applicazione è archiviata in Amazon S3.
19. In Luogo di revisione, inserisci la posizione dell'applicazione di esempio per il tuo sistema operativo e la tua regione.

    Per Amazon Linux e RHEL, istanze Amazon EC2

Region	Percorso dell'applicazione di esempio
Stati Uniti orientali (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-ea`

Region	Percorso dell'applicazione di esempio
	st-2/samples/latest/SampleA pp_Linux.zip
Stati Uniti orientali (Virginia settentrionale)	http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip
Regione Stati Uniti occidentali (California settentrionale)	http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip
Stati Uniti occidentali (Oregon)	http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip
Regione Canada (Centrale)	http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip
Regione Europa (Irlanda)	http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip
Regione Europa (Londra)	http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip
Regione Europa (Parigi)	http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip

Region	Percorso dell'applicazione di esempio
Regione Europa (Francoforte)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip`
Regione di Israele (Tel Aviv)	`https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Linux.zip`
Regione Asia Pacifico (Hong Kong)	`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`
Regione Asia Pacifico (Seoul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip`
Regione Asia Pacifico (Singapore)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip`
Regione Asia Pacifico (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip`

Region	Percorso dell'applicazione di esempio
Regione Asia Pacifico (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Linux.zip`
Regione Asia Pacifico (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip`
Regione Sud America (San Paolo)	`http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip`

Per istanze Windows Server Amazon EC2

Region	Percorso dell'applicazione di esempio
Stati Uniti orientali (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip`
Stati Uniti orientali (Virginia settentrionale)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip`
Regione Stati Uniti occidentali (California settentrionale)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip`

Region	Percorso dell'applicazione di esempio
Stati Uniti occidentali (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip`
Regione Canada (Centrale)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip`
Regione Europa (Irlanda)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip`
Regione Europa (Londra)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip`
Regione Europa (Parigi)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip`
Regione Europa (Francoforte)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip`
Regione di Israele (Tel Aviv)	`https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Windows.zip`

Region	Percorso dell'applicazione di esempio
Regione Asia Pacifico (Hong Kong)	`https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Regione Asia Pacifico (Seoul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip`
Regione Asia Pacifico (Singapore)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip`
Regione Asia Pacifico (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip`
Regione Asia Pacifico (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Regione Asia Pacifico (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip`
Regione Sud America (San Paolo)	`http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip`

Per le istanze di Ubuntu Server Amazon EC2

Digita la posizione della revisione dell'applicazione personalizzata archiviata in Amazon S3.

20. Lasciare vuoto il campo Deployment description (Descrizione distribuzione).
21. Espandere Advanced (Avanzate).
22. Scegli Create deployment (Crea distribuzione).

> **Note**
>
> Se è visualizzato Failed (Non riuscito) al posto di Succeeded (Riuscito), si potrebbero provare le tecniche descritte in Monitora e risolvi i problemi della distribuzione (usando il nome dell'applicazione **`SimpleDemoApp`** e il nome del gruppo di distribuzione **`SimpleDemoDG`**).

Passaggio 3: verifica i risultati

In questo passaggio, verificherai che la **`SimpleDemoApp`** revisione sia CodeDeploy stata installata sulla singola istanza Amazon EC2 nel gruppo Auto Scaling.

Argomenti

- Per controllare i risultati (CLI)
- Per controllare i risultati (console)

Per controllare i risultati (CLI)

Innanzitutto, è necessario il DNS pubblico dell'istanza Amazon EC2.

Usa il AWS CLI per ottenere il DNS pubblico dell'istanza Amazon EC2 nel gruppo Auto Scaling chiamando il comando. describe-instances

Prima di chiamare questo comando, è necessario l'ID dell'istanza Amazon EC2. Per ottenere l'ID, chiamare describe-auto-scaling-groups per **`CodeDeployDemo-AS-Group`** come effettuato prima:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --output text
```

Ora chiamare il comando describe-instances:

```
aws ec2 describe-instances --instance-id instance-id --query
 "Reservations[0].Instances[0].PublicDnsName" --output text
```

Il valore restituito è il DNS pubblico dell'istanza Amazon EC2.

Utilizzando un browser Web, mostra la SimpleDemoApp revisione distribuita su quell'istanza Amazon EC2, utilizzando un URL come il seguente:

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

Se vedi la pagina delle congratulazioni, significa che hai implementato CodeDeploy con successo una revisione su una singola istanza Amazon EC2 in un gruppo di Auto Scaling!

Successivamente, aggiungerai un'istanza Amazon EC2 al gruppo Auto Scaling. Dopo che Amazon EC2 Auto Scaling avrà aggiunto l'istanza Amazon EC2, distribuirà la revisione sulla nuova CodeDeploy istanza.

Per controllare i risultati (console)

Innanzitutto, è necessario il DNS pubblico dell'istanza Amazon EC2.

Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.

Nel pannello di navigazione di Amazon EC2, in Auto Scaling, scegli Auto Scaling Groups, quindi scegli la voce. **CodeDeployDemo-AS-Group**

Nella scheda Istanze, scegli l'ID dell'istanza Amazon EC2 nell'elenco.

Nella pagina Instances (Istanze) nella scheda Description (Descrizione) annotare il valore Public DNS (DNS pubblico). Dovrebbe essere simile a quanto segue: **ec2-01-234-567-890.compute-1.amazonaws.com**.

Utilizzando un browser Web, mostra la SimpleDemoApp revisione distribuita su quell'istanza Amazon EC2, utilizzando un URL come il seguente:

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

Se vedi la pagina delle congratulazioni, significa che hai implementato CodeDeploy con successo una revisione su una singola istanza Amazon EC2 in un gruppo di Auto Scaling!

Successivamente, aggiungi un'istanza Amazon EC2 al gruppo Auto Scaling. Dopo che Amazon EC2 Auto Scaling avrà aggiunto l'istanza Amazon EC2, distribuirà la revisione nella nuova CodeDeploy istanza Amazon EC2.

Fase 4: Aumentare il numero di istanze Amazon EC2 nel gruppo Auto Scaling

In questo passaggio, istruisci il gruppo Auto Scaling a creare un'istanza Amazon EC2 aggiuntiva. Dopo che Amazon EC2 Auto Scaling ha creato l' CodeDeploy istanza, distribuisce la revisione su di essa.

Argomenti

- Per ridimensionare il numero di istanze Amazon EC2 nel gruppo Auto Scaling (CLI)
- Per ridimensionare il numero di istanze Amazon EC2 nel gruppo di distribuzione (console)

Per ridimensionare il numero di istanze Amazon EC2 nel gruppo Auto Scaling (CLI)

1. Chiama il update-auto-scaling-group comando per aumentare da una a due le istanze Amazon EC2 nel gruppo Auto Scaling denominato**`CodeDeployDemo-AS-Group`**.

   Su macchine Linux, macOS o Unix locali:

   ```
   aws autoscaling update-auto-scaling-group \
     --auto-scaling-group-name CodeDeployDemo-AS-Group \
     --min-size 2 \
     --max-size 2 \
     --desired-capacity 2
   ```

   Nei computer Windows locali:

   ```
   aws autoscaling update-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-
   AS-Group --min-size 2 --max-size 2 --desired-capacity 2
   ```

2. Assicurati che il gruppo Auto Scaling disponga ora di due istanze Amazon EC2. Chiamare il comando describe-auto-scaling-groups per **`CodeDeployDemo-AS-Group`**:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
 CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,
 LifecycleState]" --output text
```

Non procedere finché entrambi i valori restituiti non mostrano `Healthy` e `InService`.

Per ridimensionare il numero di istanze Amazon EC2 nel gruppo di distribuzione (console)

1. Nella barra di navigazione di Amazon EC2, in Auto Scaling, scegli Auto Scaling Groups, quindi scegli. **`CodeDeployDemo-AS-Group`**
2. Seleziona Azioni, quindi scegli Modifica.
3. Nella scheda Details (Dettagli) nelle caselle Desired (Desiderato), Min (Minimo) e Max (Massimo), digitare **2** e scegliere Save (Salva).
4. Selezionare la scheda Instances (Istanze). La nuova istanza Amazon EC2 dovrebbe apparire nell'elenco. Se l'istanza non viene visualizzata, potrebbe essere necessario scegliere più volte il pulsante Refresh (Aggiorna). Non procedere finché non InServiceviene visualizzato il valore di nella colonna Lifecycle e il valore di Healthy nella colonna Health Status.

Fase 5: Controlla nuovamente i risultati

In questo passaggio, controllerai se è stata CodeDeploy installata la SimpleDemoApp revisione sulla nuova istanza nel gruppo Auto Scaling.

Argomenti

- Per verificare i risultati della distribuzione automatica (CLI)
- Per verificare i risultati della distribuzione automatica (console)

Per verificare i risultati della distribuzione automatica (CLI)

1. Prima di chiamare il comando get-deployment, è necessario l'ID dell'istanza della distribuzione automatica. Per ottenere l'ID, chiamare il comando list-deployments per l'applicazione denominata **`SimpleDemoApp`** e il gruppo di distribuzione con nome **`SimpleDemoDG`**:

```
aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-
name SimpleDemoDG --query "deployments" --output text
```

Sono presenti due ID di distribuzione. Utilizzare quello che ancora non è stato utilizzato in una chiamata al comando get-deployment:

```
aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.
[status, creator]" --output text
```

Oltre allo stato della distribuzione, dovresti vederlo `autoScaling` nell'output del comando. (`autoScaling`significa che Amazon EC2 Auto Scaling ha creato la distribuzione.)

Non procedere finché lo stato della distribuzione non mostra `Succeeded`.

2. Prima di chiamare il describe-instances comando, è necessario l'ID della nuova istanza Amazon EC2. Per ottenere l'ID, effettuare un'altra chiamata al comando describe-auto-scaling-groups per **`CodeDeployDemo-AS-Group`**:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
 CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --
output text
```

Ora effettuare una chiamata al comando describe-instances:

```
aws ec2 describe-instances --instance-id instance-id --query
 "Reservations[0].Instances[0].PublicDnsName" --output text
```

Nell'output del describe-instances comando, annota il DNS pubblico per la nuova istanza Amazon EC2.

3. Utilizzando un browser Web, mostra la `SimpleDemoApp` revisione distribuita su quell'istanza Amazon EC2, utilizzando un URL come il seguente:

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

Se viene visualizzata la pagina delle congratulazioni, significa che CodeDeploy in passato hai distribuito una revisione su un'istanza Amazon EC2 scalata in un gruppo Auto Scaling!

Per verificare i risultati della distribuzione automatica (console)

1. Accedi a e apri la console all'indirizzo. Console di gestione AWS CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy, quindi scegli Deployments.

3. Scegli l'ID di distribuzione della distribuzione creata da Amazon EC2 Auto Scaling.

   .

4. Nella pagina Deployment (Distribuzione) vengono visualizzate le informazioni sulla distribuzione. Normalmente, dovresti creare una distribuzione da solo, ma Amazon EC2 Auto Scaling ne ha creata una per tuo conto per distribuire la tua revisione nella nuova istanza Amazon EC2.
5. Quando Succeeded (Riuscito) viene visualizzato nella parte superiore della pagina, verificare i risultati dell'istanza. In primo luogo, è necessario ottenere il DNS dell'istanza:
6. Nel pannello di navigazione di Amazon EC2, in Auto Scaling, scegli Auto Scaling Groups, quindi scegli la voce. **`CodeDeployDemo-AS-Group`**
7. Nella scheda Istanze, scegli l'ID della nuova istanza Amazon EC2.
8. Nella pagina Instances (Istanze) nella scheda Description (Descrizione) annotare il valore Public DNS (DNS pubblico). Dovrebbe essere simile a quanto segue: **`ec2-01-234-567-890.compute-1.amazonaws.com`**.

Mostrare la revisione `SimpleDemoApp` distribuita all'istanza utilizzando un URL, come il seguente:

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

Se viene visualizzata la pagina delle congratulazioni, significa che CodeDeploy in passato hai distribuito una revisione su un'istanza Amazon EC2 scalata in un gruppo Auto Scaling!

Fase 6: pulizia

In questo passaggio, eliminerai il gruppo Auto Scaling per evitare addebiti continui per le risorse utilizzate durante questo tutorial. Facoltativamente, è possibile eliminare i record dei componenti di configurazione CodeDeploy e distribuzione di Auto Scaling.

Argomenti

- Per cancellare le risorse (CLI)
- Per cancellare le risorse (console)

Per cancellare le risorse (CLI)

1. Eliminare il gruppo Auto Scaling chiamando il delete-auto-scaling-group comando against. **`CodeDeployDemo-AS-Group`** Ciò terminerà anche le istanze Amazon EC2.

   ```
   aws autoscaling delete-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-AS-Group --force-delete
   ```

2. Facoltativamente, elimina il modello di avvio di Auto Scaling chiamando delete-launch-template il comando sulla configurazione di avvio denominata: **`CodeDeployDemo-AS-Launch-Template`**

   ```
   aws ec2 delete-launch-template --launch-template-name CodeDeployDemo-AS-Launch-Template
   ```

3. Facoltativamente, elimina l'applicazione da CodeDeploy chiamando il delete-application comando sull'applicazione denominata. **`SimpleDemoApp`** Così facendo, saranno eliminati tutti i record associati alla revisione, al gruppo di distribuzione e alla distribuzione.

   ```
   aws deploy delete-application --application-name SimpleDemoApp
   ```

4. Per eliminare l'associazione Systems Manager State Manager, chiamare il delete-association comando.

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

   È possibile ottenerla *`association-id`* chiamando il describe-association comando.

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

Per cancellare le risorse (console)

Per eliminare il gruppo Auto Scaling, che termina anche le istanze Amazon EC2:

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Nel pannello di navigazione di Amazon EC2, in Auto Scaling, scegli Auto Scaling Groups, quindi scegli la voce. **`CodeDeployDemo-AS-Group`**
3. Selezionare Actions (Azioni), Delete (Elimina) e Yes, Delete (Sì, elimina).

(Facoltativo) Per eliminare il modello di lancio:

1. Nella barra di navigazione, in Auto Scaling, scegli Avvia configurazioni, quindi scegli. **`CodeDeployDemo-AS-Launch-Template`**
2. Selezionare Actions (Azioni), Delete launch configuration (Elimina configurazione di avvio) e Yes, Delete (Sì, elimina).

1. Facoltativamente, elimina l'applicazione da. CodeDeploy Così facendo, saranno eliminati tutti i record associati alla revisione, al gruppo di distribuzione e alla distribuzione. Apri la console CodeDeploy all'indirizzo https://console.aws.amazon.com/codedeploy.
2. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

   Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.

3. Nell'elenco delle applicazioni, scegliere SimpleDemoApp.

4. Nella pagina Application details (Dettagli applicazione), scegliere Delete application (Elimina applicazione).
5. Quando richiesto, digitare **`Delete`**, quindi scegliere Delete (Elimina).

Per eliminare l'associazione Systems Manager State Manager:

1. Aprire la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager.
2. Nel riquadro di navigazione, seleziona State Manager.
3. Scegli l'associazione creata e seleziona Elimina.

Tutorial: utilizzare CodeDeploy per distribuire un'applicazione da GitHub

In questo tutorial, utilizzerai CodeDeploy per distribuire una revisione di un'applicazione di esempio da GitHub una singola istanza Amazon EC2 che esegue Amazon Linux, una singola istanza Red Hat Enterprise Linux (RHEL) o una singola istanza di Windows Server. Per informazioni sull' GitHub integrazione con, consulta. CodeDeploy Integrazione con CodeDeploy GitHub

> **Note**
>
> È inoltre possibile utilizzare CodeDeploy per distribuire una revisione dell'applicazione da un'istanza GitHub di Ubuntu Server. È possibile utilizzare la revisione di esempio descritta Passaggio 2: creare una revisione di esempio dell'applicazione in Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux) oppure creare una revisione compatibile con un'istanza di Ubuntu Server e. CodeDeploy Per creare una tua revisione, consulta Pianifica una revisione per CodeDeploy e Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy.

Argomenti

- Prerequisiti
- Fase 1: Configura un account GitHub
- Passaggio 2: crea un GitHub repository

- Passaggio 3: carica un'applicazione di esempio nel tuo GitHub repository
- Fase 4: Eseguire il provisioning di un'istanza
- Passaggio 5: creare un'applicazione e un gruppo di distribuzione
- Fase 6: Distribuire l'applicazione sull'istanza
- Fase 7: Monitoraggio e verifica della distribuzione
- Fase 8: Pulizia

Prerequisiti

Prima di iniziare il tutorial effettua le seguenti operazioni:

- Installa Git sul computer locale. Per installare Git, vedi Git downloads.
- Completa i passi in Guida introduttiva con CodeDeploy, inclusi l'installazione e la configurazione dell' AWS CLI. Ciò è particolarmente importante se desideri utilizzare il AWS CLI per distribuire una revisione dall' GitHub istanza.

Fase 1: Configura un account GitHub

È necessario un GitHub account per creare un GitHub repository in cui archiviare la revisione. Se hai già un GitHub account, vai avanti a. Passaggio 2: crea un GitHub repository

1. Passa a https://github.com/join.
2. Digitare un nome utente, un indirizzo e-mail e una password.
3. Scegli Iscriviti a GitHub, quindi segui le istruzioni.

Passaggio 2: crea un GitHub repository

Avrai bisogno di un GitHub repository per archiviare la revisione.

Se hai già un GitHub repository, assicurati di sostituirlo con il nome **`CodeDeployGitHubDemo`** durante questo tutorial, e poi vai avanti a. Passaggio 3: carica un'applicazione di esempio nel tuo GitHub repository

1. Nella GitHub home page, effettuate una delle seguenti operazioni:
   - In Your repositories (I tuoi repository), scegliere New repository (Nuovo repository).

- Nella barra di navigazione, scegliere Create new (Crea nuovo) (+) e quindi selezionare New repository (Nuovo repository).

2. Nella pagina Create a new repository (Crea un nuovo repository), eseguire queste operazioni:

   - Nel riquadro Repository name (Nome repository) immetti **`CodeDeployGitHubDemo`**.
   - Selezionare Public (Pubblico).

   > **Note**
   >
   > Selezionando l'opzione predefinita Public (Pubblico) chiunque può vedere questo repository. È possibile selezionare l'opzione Private (Privato) per limitare chi può visualizzare e eseguire il commit nel repository.

   - Deselezionare la casella di controllo Initialize this repository with a README (Inizializza il repository con un README). Un file `README.md` potrà essere creato manualmente al prossimo passaggio.
   - Scegli Create repository (Crea repository).

3. Seguire le istruzioni per il tipo di computer locale per utilizzare la riga di comando per creare il repository.

   > **Note**
   >
   > Se hai abilitato l'autenticazione a due fattori GitHub, assicurati di inserire il token di accesso personale anziché la password di GitHub accesso se ti viene richiesta una password. Per informazioni, consulta Fornire il codice di autenticazione 2FA.

Su macchine Linux, macOS o Unix locali:

1. Dal terminale, esegui i seguenti comandi, uno alla volta, *`user-name`* dov'è il tuo nome GitHub utente:

   ```
   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. Lasciare il terminale aperto nel percorso `/tmp/CodeDeployGitHubDemo`.

Nei computer Windows locali:

1. Dal prompt dei comandi in esecuzione come amministratore, eseguire i comandi seguenti, uno alla volta:

```
mkdir c:\temp\CodeDeployGitHubDemo
```

```
cd c:\temp\CodeDeployGitHubDemo
```

```
notepad README.md
```

2. Salvare il file `README.md` in Blocco note. Chiudere Blocco note. Esegui i seguenti comandi, uno alla volta, *`user-name`* dov'è il tuo nome GitHub utente:

```
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. Lasciare il prompt dei comandi aperto nel percorso `c:\temp\CodeDeployGitHubDemo`.

Passaggio 3: carica un'applicazione di esempio nel tuo GitHub repository

In questo passaggio, copierai una revisione di esempio da un bucket Amazon S3 pubblico nel tuo repository. GitHub Per semplicità, le revisioni di esempio fornite per questo tutorial sono singole pagine Web.

> **Note**
>
> Se utilizzi una delle tue revisioni anziché la nostra revisione di esempio, la tua revisione deve:
>
> - Seguire le linee guida in Pianifica una revisione per CodeDeploy e Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy.
> - Utilizzare il tipo di istanza corrispondente.
> - Sii accessibile dalla tua dashboard. GitHub
>
> Se la tua revisione soddisfa questi requisiti, passa alla Passaggio 5: creare un'applicazione e un gruppo di distribuzione.
> Se stai distribuendo su un'istanza di Ubuntu Server, dovrai caricare nel tuo GitHub repository una revisione compatibile con un'istanza di Ubuntu Server e. CodeDeploy Per ulteriori informazioni, consultare Pianifica una revisione per CodeDeploy e Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy.

Argomenti

- Invia una revisione di esempio da un computer Linux, macOS o Unix locale
- Eseguire il push di una revisione di esempio da un computer Windows locale

Invia una revisione di esempio da un computer Linux, macOS o Unix locale

Con il tuo terminale ancora aperto su un percorso, ad esempio `/tmp/CodeDeployGitHubDemo`, esegui i seguenti comandi uno alla volta:

> **Note**
>
> Se prevedi di eseguire la distribuzione su un'istanza di Windows Server, `SampleApp_Windows.zip` sostituiscila nei comandi. `SampleApp_Linux.zip`

```
(Amazon S3 copy command)
```

```
unzip SampleApp_Linux.zip
```

```
rm SampleApp_Linux.zip
```

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

*(Amazon S3 copy command)*Dov'è uno dei seguenti:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip . --region us-east-2` per la Regione Stati Uniti orientali (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip . --region us-east-1` per la Regione Stati Uniti orientali (Virginia settentrionale)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip . --region us-west-1` per Regione Stati Uniti occidentali (California settentrionale)

- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip . --region us-west-2` per la Regione Stati Uniti occidentali (Oregon)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip . --region ca-central-1`per la regione Canada (Centrale)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip . --region eu-west-1`per la regione Europa (Irlanda)
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip . --region eu-west-2`per la regione Europa (Londra)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip . --region eu-west-3`per la regione Europa (Parigi)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip . --region eu-central-1`per la regione Europa (Francoforte)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Linux.zip . --region il-central-1`per la regione di Israele (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Linux.zip . --region ap-east-1`per la regione Asia Pacifico (Hong Kong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip . --region ap-northeast-1`per la regione Asia Pacifico (Tokyo)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip . --region ap-northeast-2`per la regione Asia Pacifico (Seoul)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip . --region ap-southeast-1`per la regione Asia Pacifico (Singapore)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip . --region ap-southeast-2`per la regione Asia Pacifico (Sydney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Linux.zip . --region ap-southeast-4`per la regione Asia Pacifico (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip . --region ap-south-1`per la regione Asia Pacifico (Mumbai)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip . --region sa-east-1`per la regione Sud America (San Paolo)

Eseguire il push di una revisione di esempio da un computer Windows locale

Con il prompt dei comandi ancora aperto, ad esempio nella `c:\temp\CodeDeployGitHubDemo` posizione, esegui i seguenti comandi uno alla volta:

> **Note**
>
> Se prevedi di eseguire la distribuzione su un'istanza Amazon Linux o RHEL, sostituiscila `SampleApp_Windows.zip` nei `SampleApp_Linux.zip` comandi.

```
(Amazon S3 copy command)
```

Decomprimi il contenuto del file ZIP `the` direttamente nella directory locale (ad esempio `c:\temp\CodeDeployGitHubDemo`) e non in una nuova sottodirectory.

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

*(Amazon S3 copy command)*Dov'è uno dei seguenti:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip . --region us-east-2` per la Regione Stati Uniti orientali (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip . --region us-east-1` per la Regione Stati Uniti orientali (Virginia settentrionale)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip . --region us-west-1` per Regione Stati Uniti occidentali (California settentrionale)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip . --region us-west-2` per la Regione Stati Uniti occidentali (Oregon)

- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip . --region ca-central-1`per la regione Canada (Centrale)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip . --region eu-west-1`per la regione Europa (Irlanda)
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip . --region eu-west-2`per la regione Europa (Londra)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip . --region eu-west-3`per la regione Europa (Parigi)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip . --region eu-central-1`per la regione Europa (Francoforte)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Windows.zip . --region il-central-1`per la regione di Israele (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Windows.zip . --region ap-east-1`per la regione Asia Pacifico (Hong Kong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Windows.zip . --region ap-northeast-1`per la regione Asia Pacifico (Tokyo)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip . --region ap-northeast-2`per la regione Asia Pacifico (Seoul)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip . --region ap-southeast-1`per la regione Asia Pacifico (Singapore)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip . --region ap-southeast-2`per la regione Asia Pacifico (Sydney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Windows.zip . --region ap-southeast-4`per la regione Asia Pacifico (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip . --region ap-south-1`per la regione Asia Pacifico (Mumbai)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip . --region sa-east-1`per la regione Sud America (San Paolo)

Per inviare la tua revisione a un'istanza di Ubuntu Server, copia la revisione nel repository locale, quindi chiama quanto segue:

```
git add .
git commit -m "Added Ubuntu app"
git push
```

Fase 4: Eseguire il provisioning di un'istanza

In questa fase, verrà creata o configurata l'istanza in cui vuoi distribuire l'applicazione di esempio. Puoi eseguire la distribuzione su un'istanza Amazon EC2 o un'istanza locale che esegue uno dei sistemi operativi supportati da. CodeDeploy Per informazioni, consulta Sistemi operativi supportati dall'agente CodeDeploy. (Se hai già un'istanza configurata per l'uso nelle CodeDeploy distribuzioni, vai al passaggio successivo).

Per effettuare il provisioning di un'istanza

1. Segui le istruzioni per effettuare il provisioning Avvia un'istanza Amazon EC2 (console) di un'istanza.
2. Quando avvii l'istanza, ricordati di specificare un tag nella pagina Aggiungi tag. Per i dettagli su come specificare il tag, consultaAvvia un'istanza Amazon EC2 (console).

Per verificare che l' CodeDeploy agente sia in esecuzione sull'istanza

- Segui le istruzioni riportate Verifica che l' CodeDeploy agente sia in esecuzione per verificare che l'agente sia in esecuzione.

Dopo aver effettuato correttamente il provisioning dell'istanza e verificato che l' CodeDeploy agente sia in esecuzione, vai al passaggio successivo.

Passaggio 5: creare un'applicazione e un gruppo di distribuzione

In questo passaggio, utilizzerai la CodeDeploy console o creerai un'applicazione e un gruppo di distribuzione da utilizzare per distribuire la revisione di esempio dal tuo GitHub repository. AWS CLI

Creare un'applicazione e un gruppo di distribuzioni (console)

1. Accedi a Console di gestione AWS e apri la console all' CodeDeploy indirizzo. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Scegliere Create application (Crea applicazione), quindi selezionare Custom application (Applicazione personalizzata).
4. In Application name (Nome applicazione), immettere **`CodeDeployGitHubDemo-App`**.
5. In Compute Platform, scegli. EC2/On-premises
6. Scegli Crea applicazione.
7. Nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione).
8. In Deployment group name (Nome del gruppo di distribuzione), immettere **`CodeDeployGitHubDemo-DepGrp`**.
9. In Ruolo di servizio, scegli il nome del ruolo di CodeDeploy servizio che hai creato in Crea un ruolo di servizio per CodeDeploy.
10. In Tipo di distribuzione, scegli In-place.
11. Nella configurazione dell'ambiente, a seconda del tipo di istanza che stai utilizzando, scegli le istanze o le istanze Amazon EC2. On-premises Per Key (Chiave) e Value (Valore), digitare la chiave e il valore del tag dell'istanza applicati all'istanza nell'ambito della Fase 4: Eseguire il provisioning di un'istanza.
12. In Deployment configuration (Configurazione della distribuzione), scegliere CodeDeployDefault.AllatOnce.
13. In Load Balancer (Sistema di bilanciamento del carico), deselezionare Enable load balancing (Abilita il bilanciamento del carico).
14. Espandere Advanced (Avanzate).
15. In Alarms (Allarmi), selezionare Ignore alarm configuration (Ignora configurazione allarme).
16. Selezionare Create deployment group (Crea gruppo di distribuzione) e continuare con la fase successiva.

Creare un'applicazione e un gruppo di distribuzioni (CLI)

1. Chiamare il comando create-application per creare un'applicazione in CodeDeploy denominata `CodeDeployGitHubDemo-App`:

```
aws deploy create-application --application-name CodeDeployGitHubDemo-App
```

2. Chiamare il comando create-deployment-group per creare un gruppo di distribuzioni denominato `CodeDeployGitHubDemo-DepGrp`:

   - Se esegui la distribuzione su un'istanza Amazon EC2*ec2-tag-key*, è la chiave del tag dell'istanza Amazon EC2 che è stata applicata alla tua istanza Amazon EC2 come parte di. Fase 4: Eseguire il provisioning di un'istanza
   - Se esegui la distribuzione su un'istanza Amazon EC2*ec2-tag-value*, è il valore del tag dell'istanza Amazon EC2 che è stato applicato alla tua istanza Amazon EC2 come parte di. Fase 4: Eseguire il provisioning di un'istanza
   - Se esegui la distribuzione su un'istanza locale, *on-premises-tag-key* è la chiave del tag dell'istanza locale che è stata applicata all'istanza locale come parte di. Fase 4: Eseguire il provisioning di un'istanza
   - Se stai eseguendo la distribuzione su un'istanza locale, *on-premises-tag-value* è il valore del tag dell'istanza locale che è stato applicato all'istanza locale come parte di. Fase 4: Eseguire il provisioning di un'istanza
   - *service-role-arn*è l'ARN del ruolo di servizio per il ruolo di servizio creato in Creare un ruolo di servizio per. CodeDeploy Per trovare l'ARN del ruolo del servizio, seguire le istruzioni contenute in Ottieni il ruolo di servizio ARN (CLI).

```
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**
>
> Il comando create-deployment-group fornisce supporto per la creazione di trigger che determinano l'invio di notifiche Amazon SNS agli abbonati dell'argomento su eventi specifici nelle distribuzioni e nelle istanze. Il comando supporta anche opzioni per il

rollback automatico delle distribuzioni e l'impostazione di allarmi per interrompere le distribuzioni quando vengono raggiunte le soglie di monitoraggio negli allarmi Amazon. CloudWatch I comandi per queste azioni non sono inclusi in questo tutorial.

Fase 6: Distribuire l'applicazione sull'istanza

In questo passaggio, si utilizza la CodeDeploy console o si distribuisce AWS CLI la revisione di esempio dal GitHub repository all'istanza.

Per distribuire la revisione (console)

1. Nella pagina Deployment group details (Dettagli gruppo di distribuzione), scegliere Create deployment (Crea distribuzione).
2. In Deployment group (Gruppo di distribuzione), scegliere **`CodeDeployGitHubDemo-DepGrp`**.
3. In Tipo di revisione, scegli. GitHub
4. In Connect to GitHub, effettuate una delle seguenti operazioni:

   - Per creare una connessione per CodeDeploy le applicazioni a un GitHub account, esci da una scheda separata del browser Web. GitHub In GitHub Account, inserisci un nome per identificare questa connessione, quindi scegli Connetti a GitHub. La pagina Web richiede l'autorizzazione all'interazione con GitHub l' CodeDeploy applicazione denominata. `CodeDeployGitHubDemo-App` Continuare con la fase 5.
   - Per utilizzare una connessione già creata, in GitHubAccount, selezionane il nome, quindi scegli Connetti a GitHub. Continuare con la fase 7.
   - Per creare una connessione a un altro GitHub account, esci da una scheda separata del browser web. GitHub Scegli Connetti a un altro GitHub account, quindi scegli Connetti a GitHub. Continuare con la fase 5.
5. Segui le istruzioni nella pagina di accesso per accedere con il tuo GitHub account.
6. Nella pagina Authorize application (Autorizza applicazione), scegli Authorize application (Autorizza applicazione).
7. Nella pagina CodeDeploy Crea distribuzione, in Nome archivio, inserisci il nome GitHub utente che hai usato per accedere, seguito da una barra (`/`), seguito dal nome del repository in cui hai inviato la revisione dell'applicazione (ad esempio,). ***`my-github-user-name`*`/`** **`CodeDeployGitHubDemo`**

Se non si è sicuri del valore da inserire oppure si desidera specificare un repository diverso:

a. In una scheda separata del browser Web, accedi alla dashboard. GitHub

b. In Your repositories (I tuoi repository), passa il puntatore del mouse sul nome del repository di destinazione. Viene visualizzato un tooltip che mostra il nome dell' GitHub utente o dell'organizzazione, seguito da una barra (/), seguita dal nome del repository. Inserire il valore in Repository name (Nome repository).

> **Note**
>
> Se il nome del repository di destinazione non viene visualizzato in I tuoi repository, utilizza la GitHub casella di ricerca per trovare il repository di destinazione e il nome dell'utente o dell'organizzazione. GitHub

8. Nella casella Commit ID, inserisci l'ID del commit associato al push della revisione dell'applicazione a. GitHub

   In caso di dubbi sul valore da inserire:

   a. In una scheda separata del browser Web, accedi alla GitHub dashboard.

   b. In Your repositories (I tuoi repository), scegliere CodeDeployGitHubDemo.

   c. Nell'elenco dei commit, trova e copia l'ID di commit associato al push della revisione dell'applicazione su. GitHub Questo ID è in genere lungo 40 caratteri ed è composto da lettere e numeri. Non utilizzare la versione breve dell'ID commit che in genere è costituita dai primi 10 caratteri della versione più lunga.

   d. Incolla l'ID commit nella casella Commit ID (ID commit).

9. Selezionare Deploy (Distribuisci) e continuare con la fase successiva.

Per distribuire la revisione (CLI)

Prima di poter richiamare qualsiasi AWS CLI comando con cui interagisci GitHub (ad esempio il create-deployment comando, che chiamerai in seguito), devi CodeDeploy autorizzare l'uso del tuo account GitHub utente con cui interagire GitHub per l'`CodeDeployGitHubDemo-App`applicazione. Al momento, è necessario utilizzare la console CodeDeploy per eseguire questa operazione.

1. Accedi Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Scegli CodeDeployGitHubDemo-App.
4. Nella scheda Deployments (Distribuzioni), scegli Create deployment (Crea distribuzione).

   > **Note**
   >
   > Non verrà creata una nuova distribuzione. Attualmente questo è l'unico modo per CodeDeploy autorizzare l'interazione per GitHub conto del tuo account GitHub utente.

5. Dal gruppo Deployment, scegli CodeDeployGitHubDemo-DepGrp.
6. In Tipo di revisione, scegli GitHub.
7. In Connect to GitHub, effettuate una delle seguenti operazioni:

   - Per creare una connessione per CodeDeploy le applicazioni a un GitHub account, esci da una scheda separata del browser Web. GitHub In GitHub Account, digita un nome per identificare la connessione, quindi scegli Connetti a GitHub. La pagina Web richiede l'autorizzazione all'interazione con CodeDeploy l'applicazione GitHub denominata. `CodeDeployGitHubDemo-App` Continuare con la fase 8.
   - Per utilizzare una connessione già creata, in GitHubAccount, selezionane il nome, quindi scegli Connetti a GitHub. Continuare con la fase 10.
   - Per creare una connessione a un altro GitHub account, esci da una scheda separata del browser web. GitHub Scegli Connetti a un altro GitHub account, quindi scegli Connetti a GitHub. Continuare con la fase 8.

8. Segui le istruzioni nella pagina di accesso per accedere con il tuo nome GitHub utente o e-mail e password.
9. Nella pagina Authorize application (Autorizza applicazione), scegli Authorize application (Autorizza applicazione).
10. Nella pagina CodeDeploy Crea distribuzione, scegli Annulla.

11. Chiama il create-deployment comando per distribuire la revisione dal tuo GitHub repository all'istanza, dove:

    - *`repository`*è il nome del tuo GitHub account, seguito da una barra forward-slash (`/`), seguito dal nome del tuo repository (), ad esempio. `CodeDeployGitHubDemo` `MyGitHubUserName/CodeDeployGitHubDemo`

      Se non si è sicuri del valore da usare oppure se si desidera specificare un repository diverso:

      1. In una scheda separata del browser web, vai alla dashboard. GitHub
      2. In Your repositories (I tuoi repository), passa il puntatore del mouse sul nome del repository di destinazione. Viene visualizzato un tooltip che mostra il nome dell' GitHub utente o dell'organizzazione, seguito da una barra (`/`), seguita dal nome del repository. Si tratta del valore da usare.

      > **Note**
      >
      > Se il nome del repository di destinazione non compare in I tuoi repository, utilizza la GitHub casella di ricerca per trovare il repository di destinazione e il nome dell'utente o dell'organizzazione corrispondente. GitHub

    - *`commit-id`*è il commit associato alla versione della revisione dell'applicazione che hai inviato al tuo repository (ad esempio,). `f835159a...528eb76f`

      In caso di dubbi sul valore da utilizzare:

      1. In una scheda separata del browser Web, accedi alla dashboard. GitHub
      2. In Your repositories (I tuoi repository), scegliere CodeDeployGitHubDemo.
      3. Nell'elenco dei commit, trova l'ID di commit associato al push della revisione dell'applicazione a. GitHub Questo ID è in genere lungo 40 caratteri ed è composto da lettere e numeri. Non utilizzare la versione breve dell'ID commit che in genere è costituita dai primi 10 caratteri della versione più lunga. Usare questo valore.

    Se stai lavorando su una macchina Linux, macOS o Unix locale:

```
aws deploy create-deployment \
  --application-name CodeDeployGitHubDemo-App \
  --deployment-config-name CodeDeployDefault.OneAtATime \
```

```
  --deployment-group-name CodeDeployGitHubDemo-DepGrp \
  --description "My GitHub deployment demo" \
  --github-location repository=repository,commitId=commit-id
```

Se si sta utilizzando un computer Windows locale:

```
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
```

Fase 7: Monitoraggio e verifica della distribuzione

In questa fase, utilizzerai la CodeDeploy console o il AWS CLI per verificare il successo della distribuzione. Puoi usare il tuo browser Web per visualizzare la pagina Web che è stata distribuita all'istanza creata o configurata.

> **Note**
>
> Se esegui la distribuzione su un'istanza di Ubuntu Server, utilizza la tua strategia di test per determinare se la revisione distribuita funziona come previsto sull'istanza, quindi vai al passaggio successivo.

Per monitorare e verificare la distribuzione (console)

1. Nel riquadro di navigazione, espandi Deploy, quindi scegli Deployments.

2. Nell'elenco delle distribuzioni, cerca la riga con un valore Application di CodeDeployGitHubDemo-Appe un valore Deployment Group pari a. CodeDeployGitHubDemo-DepGrp Se Succeeded (Riuscito) o Failed (Non riuscito) non vengono visualizzati nella colonna Status (Stato), premere ripetutamente il pulsante Refresh (Aggiorna).
3. Se Failed (Non riuscito) viene visualizzato nella colonna Status (Stato), seguire le istruzioni in Visualizza i dettagli dell'istanza (console) per risolvere i problemi della distribuzione.
4. Se Succeeded (Riuscito) viene visualizzato nella colonna Status (Stato), è possibile verificare la distribuzione tramite il browser Web. La revisione di esempio distribuisce un'unica pagina Web all'istanza. Se esegui la distribuzione su un'istanza Amazon EC2,

nel tuo browser Web, accedi `http://public-dns` all'istanza (ad esempio,). `http://ec2-01-234-567-890.compute-1.amazonaws.com`

5. Se è possibile visualizzare la pagina Web, la distribuzione è andata a buon fine. Ora che hai implementato AWS CodeDeploy con successo una revisione di GitHub, puoi passare a. Fase 8: Pulizia

Per monitorare e verificare la distribuzione (CLI)

1. Chiamare il comando list-deployments per ottenere l'ID distribuzione per l'applicazione denominata `CodeDeployGitHubDemo-App` e il gruppo di distribuzione con nome `CodeDeployGitHubDemo-DepGrp`:

```
aws deploy list-deployments --application-name CodeDeployGitHubDemo-App --
deployment-group-name CodeDeployGitHubDemo-DepGrp --query "deployments" --output
 text
```

2. Chiamare il comando get-deployment fornendo l'ID della distribuzione nell'output del comando list-deployments:

```
aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.
[status, creator]" --output text
```

3. Se viene restituito Failed (Non riuscito), seguire le istruzioni in Visualizza i dettagli dell'istanza (console) per risolvere i problemi della distribuzione.
4. Se viene restituito Succeeded (Riuscito) è possibile verificare la distribuzione tramite il browser Web. La revisione di esempio è un'unica pagina Web distribuita all'istanza. Se esegui la distribuzione su un'istanza Amazon EC2, puoi visualizzare questa pagina nel tuo browser Web accedendo `http://public-dns` all'istanza Amazon EC2 (ad esempio,). `http://ec2-01-234-567-890.compute-1.amazonaws.com`
5. Se è possibile visualizzare la pagina Web, la distribuzione è andata a buon fine. La distribuzione dal tuo repository è stata AWS CodeDeploy eseguita con successo. GitHub

Fase 8: Pulizia

Per evitare ulteriori addebiti per le risorse utilizzate durante questo tutorial, devi terminare l'istanza Amazon EC2 e le risorse associate. Facoltativamente, puoi eliminare i record del componente di

distribuzione CodeDeploy associati a questo tutorial. Se stavi usando un GitHub repository solo per questo tutorial, ora puoi anche eliminarlo.

Per eliminare un CloudFormation pila (se hai usato il CloudFormation modello per creare un'istanza Amazon EC2)

1. Accedi a Console di gestione AWS e apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Nella colonna Stacks (Stack) scegliere lo stack che inizia con `CodeDeploySampleStack`.
3. Scegli Elimina.
4. Quando viene richiesto, scegliere nuovamente Delete stack (Elimina stack). L'istanza Amazon EC2 e il profilo e il ruolo di servizio dell'istanza IAM associati vengono eliminati.

Per annullare manualmente la registrazione e cancellare un'istanza locale (se è stato eseguito il provisioning di un'istanza locale)

1. Usa il AWS CLI per chiamare il comando deregister sull'istanza locale rappresentata qui da *`your-instance-name`* e sulla regione associata da: *`your-region`*

   ```
   aws deploy deregister --instance-name your-instance-name --no-delete-iam-user --
   region your-region
   ```

2. Dall'istanza locale, chiama il comando uninstall:

   ```
   aws deploy uninstall
   ```

Per terminare manualmente un'istanza Amazon EC2 (se hai avviato manualmente un'istanza Amazon EC2)

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Nel riquadro di navigazione, in Istanze scegli Istanze.
3. Seleziona la casella accanto all'istanza Amazon EC2 che desideri terminare. Nel menu Actions (Operazioni), passare a Instance State (Stato istanza), quindi selezionare Terminate (Termina).
4. Quando richiesto, scegliere Yes, Terminate (Sì, termina).

Per eliminare i record dei componenti CodeDeploy di distribuzione

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.

3. Scegli CodeDeployGitHubDemo-App.
4. Scegli Elimina applicazione.
5. Quando richiesto, digitare **`Delete`**, quindi scegliere Delete (Elimina).

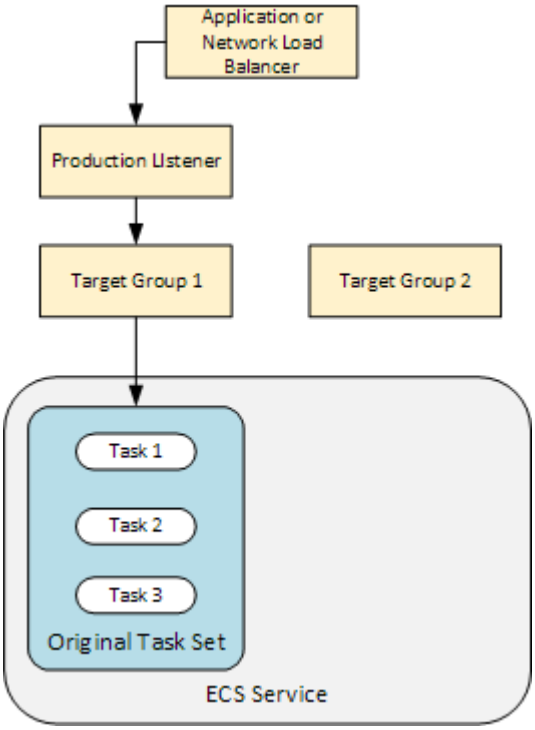
Per eliminare il tuo repository GitHub

Vedi Eliminazione di un repository nella guida. GitHub

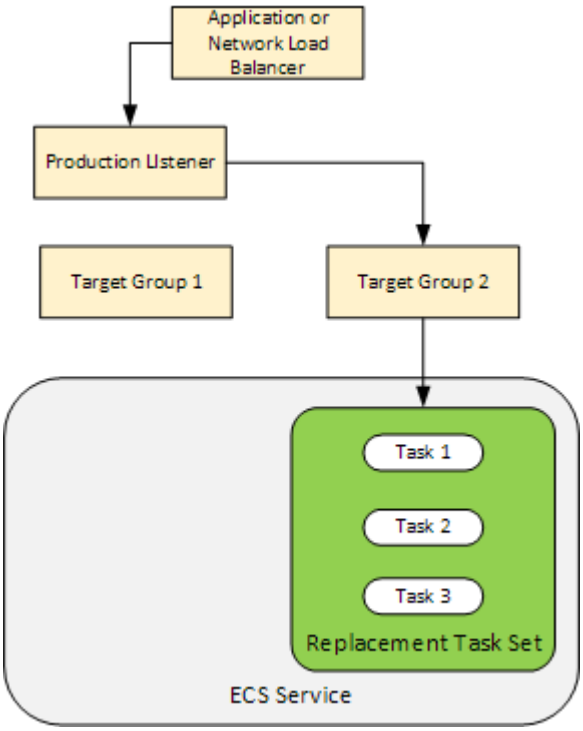
Tutorial: distribuire un'applicazione in Amazon ECS

In questo tutorial, imparerai come distribuire un'applicazione in Amazon ECS utilizzando. CodeDeploy Si inizia con un'applicazione già creata e distribuita in Amazon ECS. La prima fase consiste nell'aggiornare l'applicazione modificando il file di definizione dell'attività con un nuovo tag. Successivamente, si utilizza CodeDeploy per distribuire l'aggiornamento. Durante la distribuzione, CodeDeploy installa l'aggiornamento in un nuovo set di attività sostitutivo. Quindi, sposta il traffico di produzione dalla versione originale dell'applicazione Amazon ECS, che si trova nel set di attività originale, alla versione aggiornata nel set di attività sostitutivo.

Durante una distribuzione di Amazon ECS, CodeDeploy utilizza un sistema di bilanciamento del carico configurato con due gruppi target e un listener di traffico di produzione. Il diagramma seguente mostra come il load balancer, il listener di produzione, i gruppi target e l'applicazione Amazon ECS sono correlati prima dell'inizio della distribuzione. In questa esercitazione viene utilizzato un Application Load Balancer. Puoi anche usare un Network Load Balancer.



Una volta completata la distribuzione, il listener del traffico di produzione serve il traffico verso il nuovo set di attività di sostituzione e il set di attività originale viene terminato. Il seguente diagramma mostra il modo in cui le risorse sono correlate dopo una distribuzione riuscita. Per ulteriori informazioni, consulta Cosa succede durante una distribuzione di Amazon ECS.



Per informazioni su come utilizzare AWS CLI per distribuire un'applicazione in Amazon ECS, consulta Tutorial: Creazione di un servizio utilizzando una blue/green distribuzione. Per informazioni su come rilevare e distribuire automaticamente le modifiche CodePipeline a un servizio Amazon ECS con CodeDeploy, consulta Tutorial: Create a pipeline with an Amazon ECR source and deployment. ECS-to-CodeDeploy

Dopo aver completato questo tutorial, puoi utilizzare l' CodeDeploy applicazione e il gruppo di distribuzione che hai creato per aggiungere un test di convalida della distribuzione. Tutorial: distribuisci un servizio Amazon ECS con un test di convalida

Argomenti

- Prerequisiti
- Fase 1: aggiorna la tua applicazione Amazon ECS
- Fase 2: Creare il file AppSpec
- Passaggio 3: utilizza la CodeDeploy console per distribuire l'applicazione
- Fase 4: pulizia

Prerequisiti

Per completare questo tutorial, è necessario innanzitutto:

- Completare le fasi 2 e 3 in Guida introduttiva con CodeDeploy.
- Crea un Application Load Balancer configurato con due gruppi target e un listener. Per informazioni sulla creazione di un sistema di bilanciamento del carico tramite la console, consulta Configura un sistema di bilanciamento del carico, gruppi target e listener per le distribuzioni di CodeDeploy Amazon ECS. Per informazioni sulla creazione di un load balancer utilizzando AWS CLI, consulta Step 1: Create an Application Load Balancer nella Amazon Elastic Container Service User Guide. Quando si crea il sistema di bilanciamento del carico, annotare quanto segue per questo tutorial:
  - Il nome del sistema di bilanciamento del carico.
  - I nomi dei gruppi target.
  - La porta utilizzata dal listener del sistema di bilanciamento del carico.
- Crea un cluster e un servizio Amazon ECS. Per ulteriori informazioni, consulta i passaggi 2, 3 e 4 del Tutorial: Creazione di un servizio utilizzando una blue/green distribuzione nella Amazon Elastic Container Service User Guide. Annota quanto segue per questo tutorial:
  - Il nome del tuo cluster Amazon ECS.

  - L'ARN della definizione dell'attività utilizzata dal tuo servizio Amazon ECS.
  - Il nome del contenitore utilizzato dal tuo servizio Amazon ECS.
- Crea un bucket Amazon S3 per il tuo file. AppSpec

Fase 1: aggiorna la tua applicazione Amazon ECS

In questa sezione, aggiorni la tua applicazione Amazon ECS con una nuova revisione della definizione del task. La revisione aggiornata aggiunge una nuova coppia di chiavi e tag. NelPassaggio 3: utilizza la CodeDeploy console per distribuire l'applicazione, distribuisci la versione aggiornata della tua applicazione Amazon ECS.

Per aggiornare la definizione dell'attività

1. Apri la console all'indirizzo. https://console.aws.amazon.com/ecs/v2
2. Nel riquadro di navigazione, scegli Definizioni di attività.
3. Scegli la definizione di attività utilizzata dal tuo servizio Amazon ECS.
4. Seleziona la revisione della definizione di attività, quindi scegli Crea nuova revisione, Crea nuova revisione.
5. Per questo tutorial, è necessario effettuare un piccolo aggiornamento alla definizione dell'attività aggiungendo un tag. Nella parte inferiore della pagina, sotto Tag, crea un nuovo tag inserendo una nuova coppia di chiavi e valori.
6. Scegli Create (Crea).

   Il numero di revisione della definizione dell'attività viene incrementato di uno.
7. Scegli la scheda JSON. Prendere nota di quanto segue perché queste informazioni sono necessarie nella fase successiva.

   - Il valore per `taskDefinitionArn`. Il formato è: `arn:aws:ecs:`*`aws-region`*`:`*`account-id`*`:task-definition/`*`task-definition-family`*`:`*`task-definition-revision`*. Questo è l'ARN della definizione di attività aggiornata.
   - Nell'elemento `containerDefinitions`, il valore per `name`. Questo è il nome del container.
   - Nell'elemento `portMappings`, il valore per `containerPort`. Questa è la porta per il container.

Fase 2: Creare il file AppSpec

In questa sezione, crei il tuo AppSpec file e lo carichi nel bucket Amazon S3 che hai creato nella sezione. Prerequisiti Il AppSpec file per una distribuzione Amazon ECS specifica la definizione dell'attività, il nome del contenitore e la porta del contenitore. Per ulteriori informazioni, consultare AppSpec Esempio di file per una distribuzione Amazon ECS e AppSpec sezione «risorse» per le distribuzioni di Amazon ECS.

Per creare il tuo file AppSpec

1. Se vuoi creare il tuo AppSpec file usando YAML, crea un file denominato. `appspec.yml` Se vuoi creare il tuo AppSpec file usando JSON, crea un file denominato. `appspec.json`
2. Scegli la scheda appropriata, a seconda che utilizzi YAML o JSON per il tuo AppSpec file, e copiane il contenuto nel AppSpec file appena creato. Per la proprietà `TaskDefinition`, utilizzare l'ARN della definizione dell'attività annotato nella sezione Fase 1: aggiorna la tua applicazione Amazon ECS.

   JSON AppSpec

   ```
   {
     "version": 0.0,
     "Resources": [
       {
         "TargetService": {
           "Type": "AWS::ECS::Service",
           "Properties": {
             "TaskDefinition": "arn:aws:ecs:aws-region-id:aws-account-id:task-
   definition/ecs-demo-task-definition:revision-number",
             "LoadBalancerInfo": {
               "ContainerName": "your-container-name",
               "ContainerPort": your-container-port
             }
           }
         }
       }
     ]
   }
   ```

   YAML AppSpec

   ```
   version: 0.0
   ```

```
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:aws-region-id:aws-account-id:task-
definition/ecs-demo-task-definition:revision-number"
        LoadBalancerInfo:
          ContainerName: "your-container-name"
          ContainerPort: your-container-port
```

> **Note**
>
> Il set di attività di sostituzione eredita la sottorete, il gruppo di sicurezza, la versione della piattaforma e i valori IP pubblici assegnati dal set di attività originale. È possibile sovrascrivere questi valori per il set di attività sostitutivo impostando le relative proprietà opzionali nel file. AppSpec Per ulteriori informazioni, consultare AppSpec sezione «risorse» per le distribuzioni di Amazon ECS e AppSpec Esempio di file per una distribuzione Amazon ECS.

3. Carica il AppSpec file nel bucket S3 che hai creato come prerequisito per questo tutorial.

Passaggio 3: utilizza la CodeDeploy console per distribuire l'applicazione

In questa sezione, crei un' CodeDeploy applicazione e un gruppo di distribuzione per distribuire l'applicazione aggiornata in Amazon ECS. Durante la distribuzione, CodeDeploy sposta il traffico di produzione dell'applicazione alla nuova versione in un nuovo set di attività sostitutivo. Per completare questa fase, sono necessari i seguenti elementi:

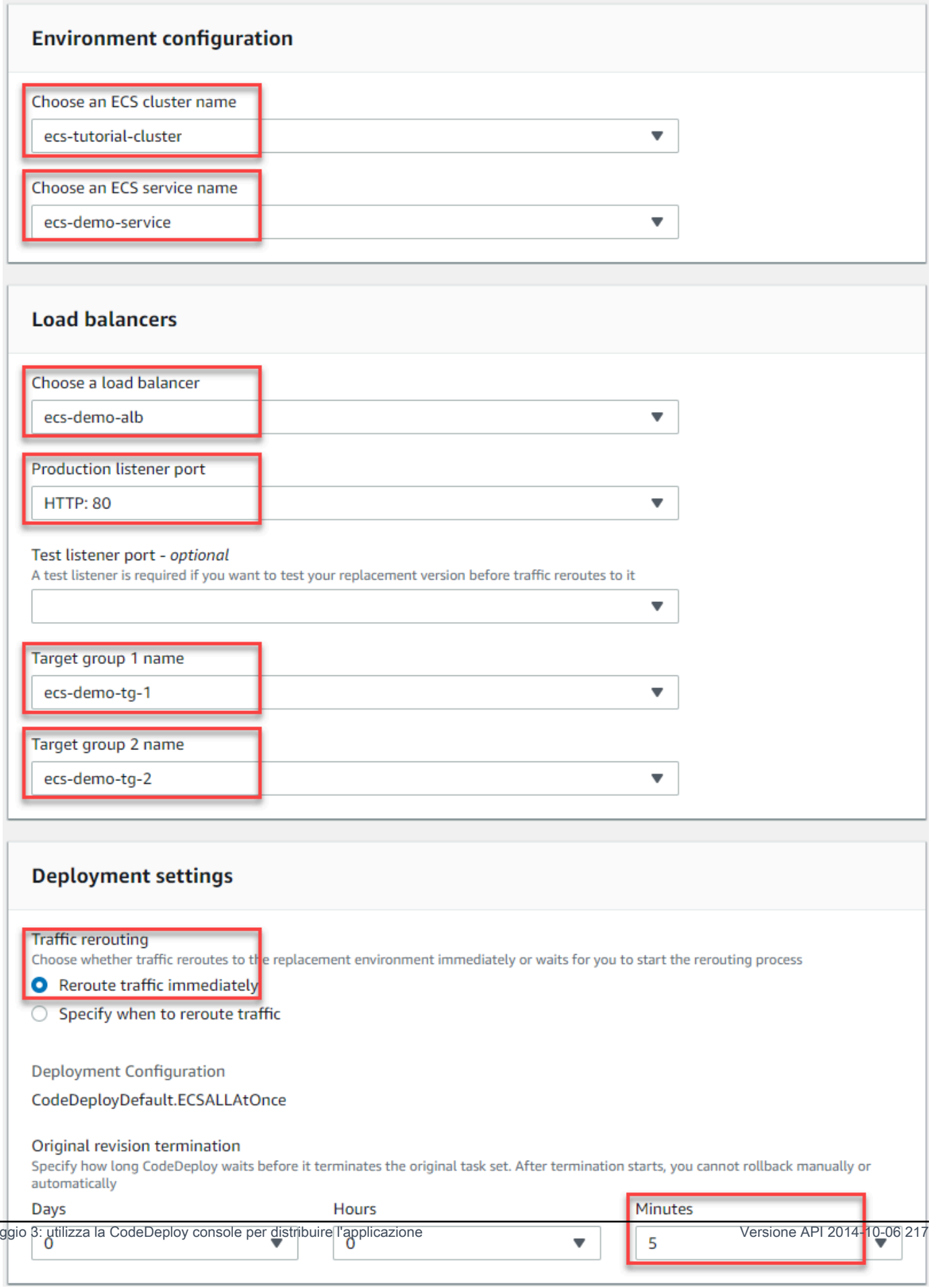
- Il nome del tuo cluster Amazon ECS.
- Il nome del tuo servizio Amazon ECS.
- Il nome dell'Application Load Balancer.
- La porta del listener di produzione.
- I nomi dei gruppi target.
- Nome del bucket S3 creato.

Per creare un'applicazione CodeDeploy

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy/.
2. Scegli Crea applicazione.
3. In Application name (Nome applicazione), immettere **ecs-demo-codedeploy-app**.
4. In Compute platform (Piattaforma di calcolo), scegli Amazon ECS.
5. Scegli Crea applicazione.

Per creare un gruppo CodeDeploy di distribuzione

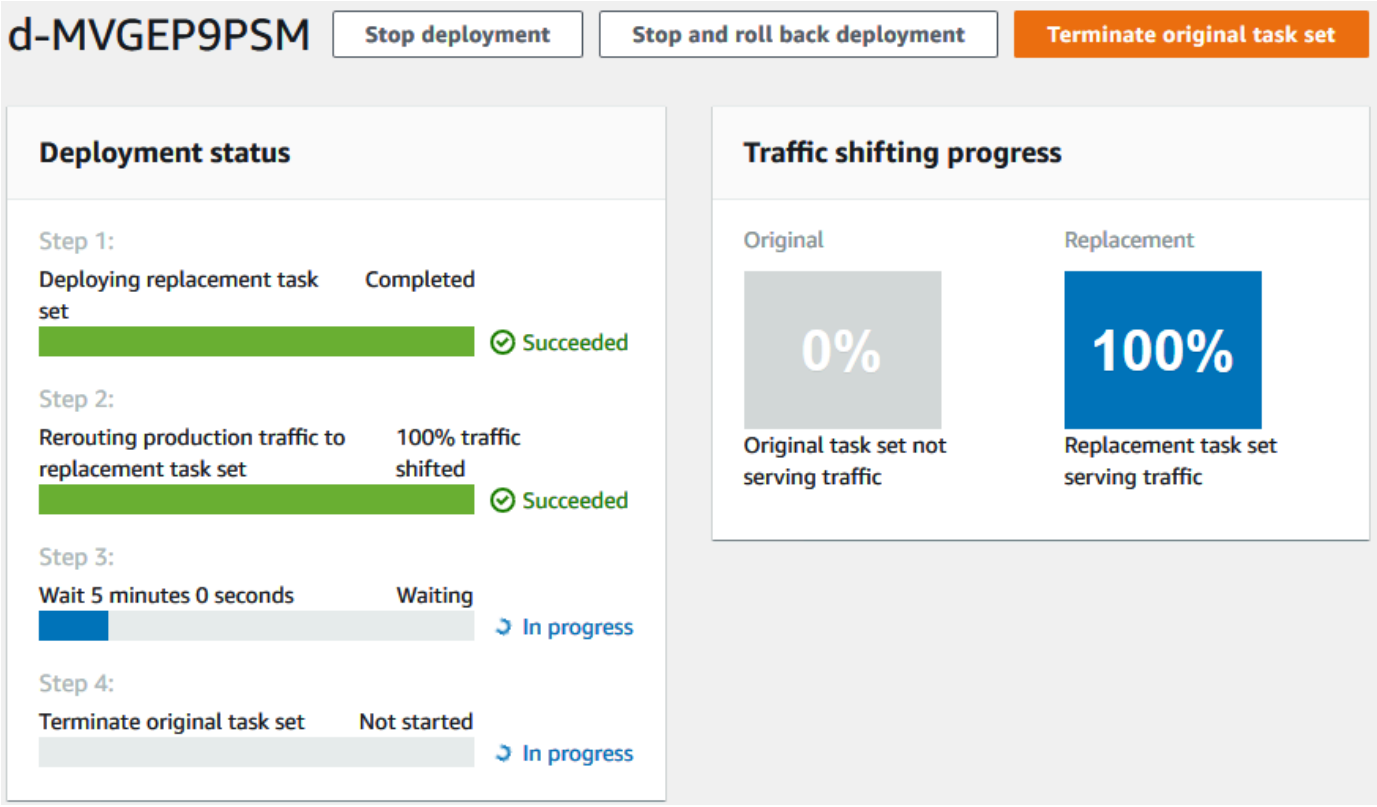
1. Nella pagina delle applicazioni, nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione).
2. In Deployment group name (Nome del gruppo di distribuzione), immettere **ecs-demo-dg**.
3. In Ruolo di servizio, scegli un ruolo di servizio che garantisca CodeDeploy l'accesso ad Amazon ECS. Per ulteriori informazioni, consulta Gestione delle identità e degli accessi per AWS CodeDeploy.
4. In Configurazione dell'ambiente, scegli il nome del cluster Amazon ECS e il nome del servizio.
5. In Load balancer, scegli il nome del load balancer che serve il traffico verso il tuo servizio Amazon ECS.
6. Da Production listener port, scegli la porta e il protocollo per il listener che invia il traffico di produzione al tuo servizio Amazon ECS (ad esempio, HTTP: 80). Questo tutorial non include un listener di test opzionale, quindi non scegliere una porta da Test listener port (Porta listener test).
7. Da Target group 1 name (Nome gruppo di destinazione 1) e Target group 2 name (Nome gruppo di destinazione 2), scegliere i due diversi gruppi di destinazione utilizzati per instradare il traffico durante la distribuzione. Assicurati che questi siano i gruppi di destinazione creati per il sistema di bilanciamento del carico. Non importa quale viene utilizzato per il gruppo target 1 e quale viene utilizzato per il gruppo target 2.
8. Scegliere Reroute traffic immediately (Reinstrada immediatamente il traffico).
9. Per Original revision termination (Terminazione revisione originale), scegliere 0 giorni, 0 ore e 5 minuti. In questo modo la distribuzione viene completata più rapidamente rispetto all'utilizzo dell'impostazione predefinita (1 ora).


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. Scegliere Create deployment group (Crea gruppo di distribuzione).

Per distribuire la tua applicazione Amazon ECS

1. Nella pagina della console del gruppo di distribuzione, scegliere Create deployment (Crea distribuzione).
2. Per Deployment group (Gruppo di distribuzione), scegliere ecs-demo-dg.
3. Per Revision type (Tipo di revisione), scegliere My application is stored in Amazon S3 (La mia applicazione è archiviata in Amazon S3). In Revision location (Posizione revisione), immettere il nome del bucket S3.
4. Per Revision file type (Tipo di file revisione), scegliere .json o .yaml, a seconda dei casi.
5. (Facoltativo) In Deployment description (Descrizione distribuzione), immetti una descrizione per questa distribuzione.
6. Scegli Create deployment (Crea distribuzione).
7. In Deployment status (Stato distribuzione), è possibile monitorare la distribuzione. Dopo che il 100% del traffico di produzione è stato indirizzato al set di attività sostitutivo e prima della scadenza del tempo di attesa di cinque minuti, puoi scegliere Termina il set di attività originale per terminare immediatamente il set di attività originale. Se non si sceglie Terminate original task set (Termina set di attività originale), il set di attività originale termina dopo la scadenza del tempo di attesa di cinque minuti specificato.



Fase 4: pulizia

Il tutorial successivo, Tutorial: distribuisci un servizio Amazon ECS con un test di convalida, si basa su questo tutorial e utilizza il gruppo di distribuzione e l'applicazione CodeDeploy creati. Per seguire le fasi del tutorial, ignorare questa fase e non eliminare le risorse create.

> **Note**
>
> Sul tuo AWS account non sono previsti addebiti per le CodeDeploy risorse che hai creato.

I nomi delle risorse in questi passaggi sono i nomi suggeriti in questo tutorial (ad esempio, **`ecs-demo-codedeploy-app`** per il nome dell' CodeDeploy applicazione). Se hai utilizzato nomi diversi, assicurati di utilizzarli durante la pulizia.

1. Utilizzate il comando delete-deployment-group per eliminare il gruppo di distribuzione. CodeDeploy

```
aws deploy delete-deployment-group --application-name ecs-demo-codedeploy-app --
deployment-group-name ecs-demo-dg --region aws-region-id
```

2. Utilizzate il comando delete-application per eliminare l'applicazione. CodeDeploy

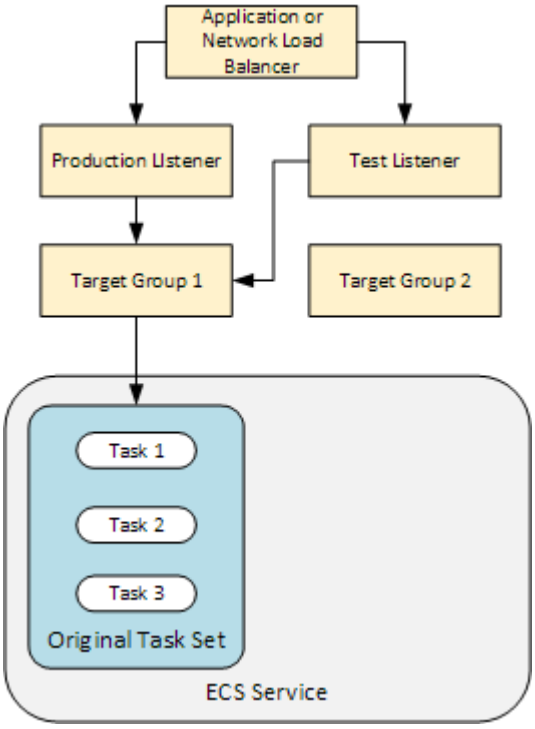
```
aws deploy delete-application --application-name ecs-demo-codedeploy-app --
region aws-region-id
```

Tutorial: distribuisci un servizio Amazon ECS con un test di convalida

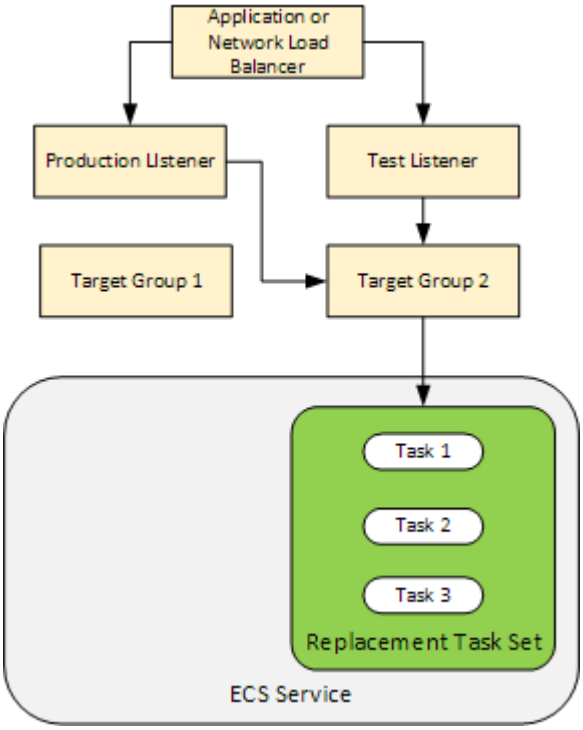
In questo tutorial, imparerai come usare una funzione Lambda per convalidare parte della distribuzione di un'applicazione Amazon ECS aggiornata. Questo tutorial utilizza l' CodeDeploy applicazione, il gruppo di CodeDeploy distribuzione e l'applicazione Amazon ECS in Tutorial: distribuire un'applicazione in Amazon ECS cui hai utilizzato. Completare il tutorial prima di iniziare questo.

Per aggiungere il test di convalida, devi prima implementare il test in una funzione Lambda. Successivamente, nel AppSpec file di distribuzione, specifichi la funzione Lambda per l'hook del ciclo di vita che desideri testare. Se un test di convalida ha esito negativo, la distribuzione viene interrotta, viene eseguito il rollback e contrassegnato come non riuscito. Se il test va a buon fine, la distribuzione continua con l'hook o l'evento del ciclo di vita della distribuzione successivo.

Durante una distribuzione Amazon ECS con test di convalida, CodeDeploy utilizza un sistema di bilanciamento del carico configurato con due gruppi target: un listener di traffico di produzione e un listener di traffico di test. Il diagramma seguente mostra come il load balancer, i listener di produzione e test, i gruppi target e l'applicazione Amazon ECS sono correlati prima dell'inizio della distribuzione. In questa esercitazione viene utilizzato un Application Load Balancer. Puoi anche usare un Network Load Balancer.



Durante una distribuzione di Amazon ECS, sono disponibili cinque hook del ciclo di vita per i test. Questo tutorial implementa un test durante il terzo hook di distribuzione del ciclo di vita, `AfterAllowTestTraffic`. Per ulteriori informazioni, consulta Elenco degli hook relativi agli eventi del ciclo di vita per una distribuzione Amazon ECS. Una volta completata la distribuzione, il listener del traffico di produzione serve il traffico verso il nuovo set di attività di sostituzione e il set di attività originale viene terminato. Il seguente diagramma mostra il modo in cui le risorse sono correlate dopo una distribuzione riuscita. Per ulteriori informazioni, consulta Cosa succede durante una distribuzione di Amazon ECS.



> **Note**
>
> Il completamento di questo tutorial potrebbe comportare addebiti sul tuo account. AWS Questi includono possibili addebiti per CodeDeploy AWS Lambda, e CloudWatch. Per ulteriori informazioni, consulta AWS CodeDeploy prezzi, AWS Lambda prezzi e CloudWatch prezzi Amazon.

Argomenti

- Prerequisiti
- Fase 1: Creare un listener di test
- Fase 2: aggiorna la tua applicazione Amazon ECS
- Fase 3: Creare una funzione Lambda dell'hook del ciclo di vita
- Passaggio 4: aggiorna il file AppSpec
- Fase 5: utilizza la CodeDeploy console per distribuire il servizio Amazon ECS
- Fase 6: Visualizza l'output della funzione Lambda hook in Logs CloudWatch
- Fase 7: pulire

Prerequisiti

Per completare questo tutorial, è necessario innanzitutto:

- Completa i prerequisiti in Prerequisiti per Tutorial: distribuire un'applicazione in Amazon ECS.
- Completa le fasi descritte in Tutorial: distribuire un'applicazione in Amazon ECS. Prendi nota di quanto segue:
  - Il nome del sistema di bilanciamento del carico.
  - I nomi dei gruppi target.
  - La porta utilizzata dal listener del sistema di bilanciamento del carico.
  - L'ARN del sistema di bilanciamento del carico. Per creare un nuovo listener si dovrà utilizzare questo.
  - L'ARN di uno dei gruppi target. Per creare un nuovo listener si dovrà utilizzare questo.
  - L' CodeDeploy applicazione e il gruppo di distribuzione che crei.
  - Il AppSpec file creato che viene utilizzato dalla CodeDeploy distribuzione. Questo file viene modificato in questo tutorial.

Fase 1: Creare un listener di test

Una distribuzione Amazon ECS con test di convalida richiede un secondo listener. Questo listener viene utilizzato per fornire traffico di test all'applicazione Amazon ECS aggiornata in un set di attività sostitutivo. I test di convalida vengono eseguiti sul traffico di test.

Il listener per il traffico di test può utilizzare uno dei gruppi target. Usa il AWS CLI comando create-listener per creare un secondo listener con una regola predefinita che inoltra il traffico di test alla porta 8080. Utilizzare l'ARN del sistema di bilanciamento del carico e l'ARN di uno dei gruppi target.

```
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
```

Fase 2: aggiorna la tua applicazione Amazon ECS

In questa sezione, aggiorni la tua applicazione Amazon ECS per utilizzare una nuova revisione della definizione del task. È possibile creare la nuova revisione e aggiungervi un aggiornamento secondario aggiungendo un tag.

Per aggiornare la definizione dell'attività

1. Apri la console Amazon ECS classic all'indirizzo https://console.aws.amazon.com/ecs/.
2. Nel riquadro di navigazione, scegli Definizioni di attività.
3. Seleziona la casella di controllo per la definizione dell'attività utilizzata dal tuo servizio Amazon ECS.
4. Scegliere Create new revision (Crea nuova revisione).
5. Effettuare un piccolo aggiornamento alla definizione dell'attività aggiungendo un tag. Nella parte inferiore della pagina, in Tags (Tag), creare un nuovo tag immettendo una nuova coppia chiave-valore.
6. Scegli Create (Crea). Il numero di revisione della definizione dell'attività è stato incrementato di uno.
7. Scegli la scheda JSON. Annotare il valore per `taskDefinitionArn`. Il formato è: `arn:aws:ecs:`*`aws-region`*`:` *`account-id`*`:task-definition/`*`task-definition-family`*`:` *`task-definition-revision`*. Questo è l'ARN della definizione di attività aggiornata.

Fase 3: Creare una funzione Lambda dell'hook del ciclo di vita

In questa sezione, implementerai una funzione Lambda per l'hook della tua distribuzione Amazon ECS. `AfterAllowTestTraffic` La funzione Lambda esegue un test di convalida prima dell'installazione dell'applicazione Amazon ECS aggiornata. Per questo tutorial, viene restituita la funzione Lambda. `Succeeded` Durante una distribuzione reale, i test di convalida restituiscono `Succeeded` o `Failed`, a seconda del risultato del test di convalida. Inoltre, durante una distribuzione nel mondo reale, puoi implementare una funzione di test Lambda per uno o più degli altri eventi del ciclo di vita della distribuzione Amazon ECS (`BeforeInstall`,, e). `AfterInstall` `BeforeAllowTraffic` `AfterAllowTraffic` Per ulteriori informazioni, consulta Elenco degli hook relativi agli eventi del ciclo di vita per una distribuzione Amazon ECS.

È necessario un ruolo IAM per creare la funzione Lambda. Il ruolo concede alla funzione Lambda il permesso di scrivere CloudWatch nei registri e impostare lo stato di un hook del ciclo di CodeDeploy vita.

Per creare un ruolo IAM

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel riquadro di navigazione selezionare Roles (Ruoli), quindi Create role (Crea ruolo).
3. Creare un ruolo con le seguenti proprietà:

   - Trusted entity (Entità attendibile): AWS Lambda.
   - Autorizzazioni:. AWSLambdaBasicExecutionRole Ciò concede alla funzione Lambda il permesso di scrivere CloudWatch nei registri.
   - Nome ruolo: **`lambda-cli-hook-role`**.

   Per ulteriori informazioni, consulta Creare un AWS Lambda ruolo di esecuzione.
4. Collega l'autorizzazione `codedeploy:PutLifecycleEventHookExecutionStatus` al ruolo creato. Ciò concede alle funzioni Lambda l'autorizzazione a impostare lo stato di CodeDeploy un hook del ciclo di vita durante una distribuzione. Per ulteriori informazioni, consulta Aggiungere i permessi di identità IAM nella Guida per l'AWS Identity and Access Management utente e PutLifecycleEventHookExecutionStatusnel riferimento API. CodeDeploy

Per creare una funzione Lambda a forma di **`AfterAllowTestTraffic`**hook

1. Crea un file denominato `AfterAllowTestTraffic.js` con i seguenti contenuti.

```
'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. Crea un pacchetto di distribuzione Lambda.

```
zip AfterAllowTestTraffic.zip AfterAllowTestTraffic.js
```

3. Usa il `create-function` comando per creare una funzione Lambda per il tuo `AfterAllowTestTraffic` hook.

```
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. Prendi nota dell'ARN della tua funzione Lambda nella risposta. `create-function` Questo ARN viene utilizzato quando si aggiorna il AppSpec file della CodeDeploy distribuzione nel passaggio successivo.

Passaggio 4: aggiorna il file AppSpec

In questa sezione, aggiorni il AppSpec file con una `Hooks` sezione. Nella `Hooks` sezione, si specifica una funzione Lambda per il `AfterAllowTestTraffic` lifecycle hook.

Per aggiornare il file AppSpec

1. Apri il AppSpec file che hai creato in Fase 2: Creare il file AppSpecTutorial: distribuire un'applicazione in Amazon ECS.
2. Aggiornare la proprietà `TaskDefinition` con l'ARN della definizione dell'attività annotato in Fase 2: aggiorna la tua applicazione Amazon ECS.
3. Copia e incolla la `Hooks` sezione nel tuo AppSpec file di file. Aggiorna l'ARN dopo `AfterAllowTestTraffic` con l'ARN della funzione Lambda che hai annotato. Fase 3: Creare una funzione Lambda dell'hook del ciclo di vita

   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. Salva il AppSpec file e caricalo nel relativo bucket S3.

Fase 5: utilizza la CodeDeploy console per distribuire il servizio Amazon ECS

In questa sezione, aggiornerai il gruppo di distribuzione specificando la porta per il listener di test. Questo è il listener creato in Fase 1: Creare un listener di test. Nel corso della distribuzione, CodeDeploy esegue il test di convalida durante l'hook del ciclo di vita della distribuzione `AfterAllowTestTraffic` utilizzando il traffico di test servito al set di attività di sostituzione mediante il listener di test. Il test di convalida restituisce il risultato `Succeeded`, quindi la distribuzione procede con il successivo evento del ciclo di vita della distribuzione. In uno scenario reale, la funzione di test restituisce `Succeeded` o `Failed`.

Per aggiungere un listener di test al gruppo di distribuzione

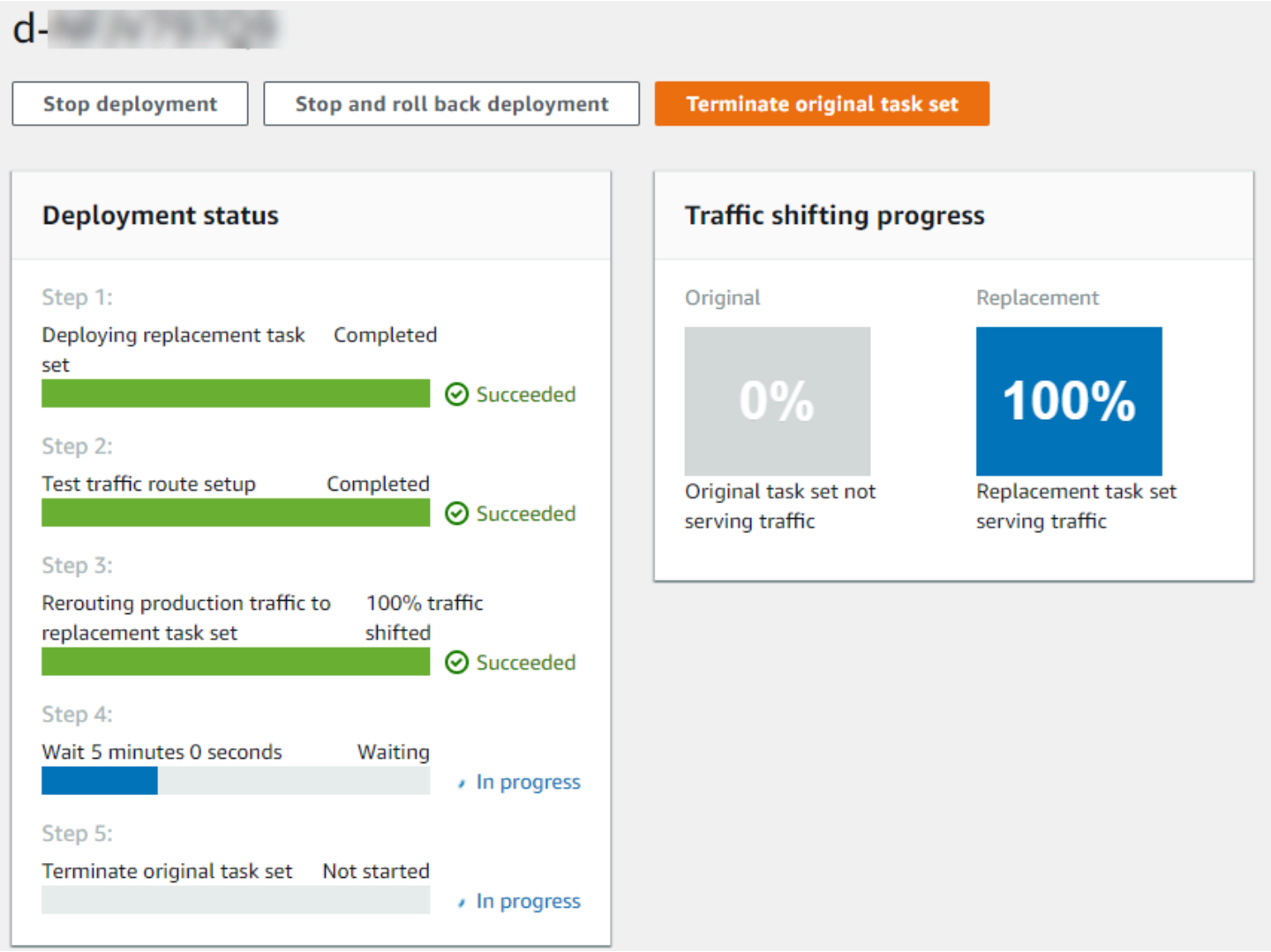
1. Accedi Console di gestione AWS e apri la CodeDeploy console all'indirizzo. https://console.aws.amazon.com/codedeploy/
2. Nel riquadro di navigazione , scegliere Applications (Applicazioni).
3. Scegliere l'applicazione creata in Tutorial: distribuire un'applicazione in Amazon ECS. Se hai utilizzato il nome suggerito, è ecs-demo-codedeploy-app.

4. In Deployment group (Gruppo di distribuzione), scegliere il gruppo di distribuzione appena creato in Tutorial: distribuire un'applicazione in Amazon ECS. Se hai utilizzato il nome suggerito, è ecs-demo-dg.
5. Scegli Modifica.
6. Da Test listener port (Porta listener di test), scegliere la porta e il protocollo per il listener di test creato in precedenza in questo tutorial. Deve essere HTTP: 8080.
7. Scegli Save changes (Salva modifiche).

Per distribuire la tua applicazione Amazon ECS

1. Nella pagina della console del gruppo di distribuzione, scegliere Create deployment (Crea distribuzione).
2. Per Deployment group (Gruppo di distribuzione), scegliere ecs-demo-dg.
3. Per Revision type (Tipo di revisione), scegliere My application is stored in Amazon S3 (La mia applicazione è archiviata in Amazon S3). In Posizione di revisione, inserisci il nome del bucket e del AppSpec file S3 (ad esempio,). **`s3://my-s3-bucket/appspec.json`**
4. Per Revision file type (Tipo di file revisione), scegliere .json o .yaml a seconda dei casi.
5. (Facoltativo) In Deployment description (Descrizione distribuzione), immetti una descrizione per questa distribuzione.
6. Scegli Create deployment (Crea distribuzione).

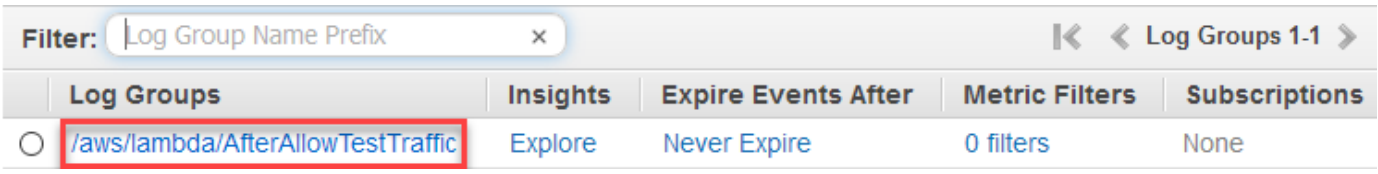
Puoi monitorare la distribuzione in Deployment status (Stato distribuzione). Dopo che il 100% del traffico di produzione è stato indirizzato al set di attività sostitutivo, puoi scegliere Termina il set di attività originale per terminare immediatamente il set di attività originale. Se non si sceglie Terminate original task set (Termina set di attività originale), il set di attività originale termina dopo la durata specificata al momento della creazione del gruppo di distribuzione.



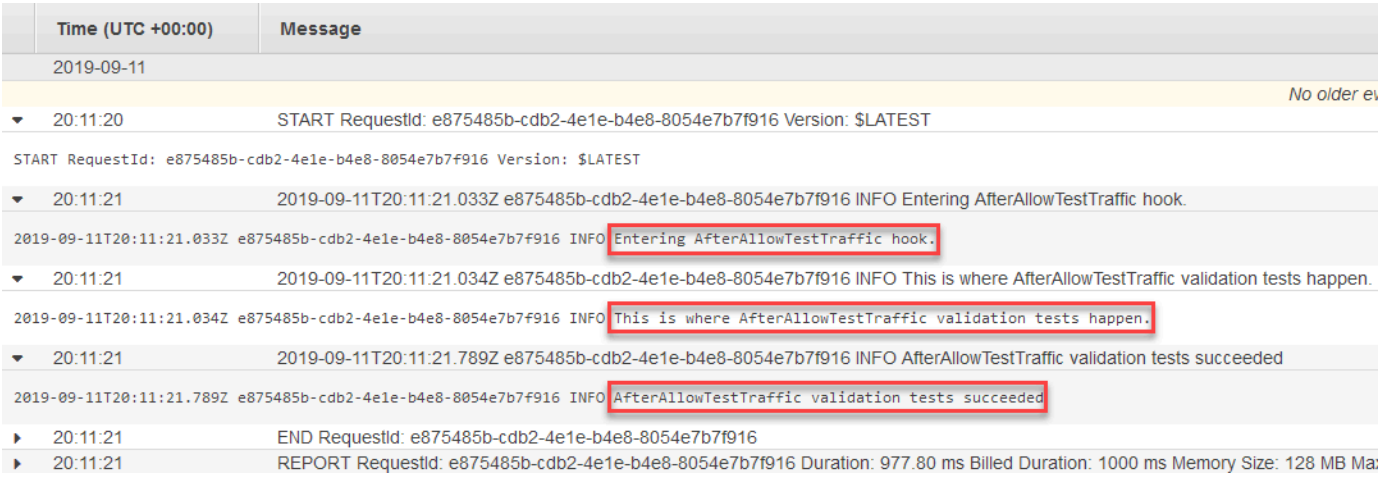
Fase 6: Visualizza l'output della funzione Lambda hook in Logs CloudWatch

Se l' CodeDeploy implementazione ha esito positivo, anche i test di convalida delle funzioni hook Lambda hanno esito positivo. Puoi confermarlo esaminando il registro della funzione hook in Logs. CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione scegliere Log. Dovresti vedere un nuovo gruppo di log per la funzione hook Lambda che hai specificato nel tuo AppSpec file.



3. Scegliere il nuovo gruppo di log. Dovrebbe essere/aws/lambda/AfterAllowTestTrafficHook.
4. Scegli il flusso di log. Se si visualizza più di un flusso di log, scegliere quello con la data e l'ora più recenti in Last Event Time (Ora ultimo evento).
5. Espandi gli eventi del flusso di registro per confermare che la funzione Lambda hook abbia scritto messaggi di successo nel registro. Quanto segue mostra che la funzione `AfterAllowTraffic` Lambda hook ha avuto successo.



Fase 7: pulire

Una volta terminato questo tutorial, rimuovi le risorse associate per evitare costi aggiuntivi per risorse che non utilizzi. I nomi delle risorse in questo passaggio sono i nomi suggeriti in questo tutorial (ad esempio, **`ecs-demo-codedeploy-app`** per il nome dell' CodeDeploy applicazione). Se sono stati utilizzati nomi diversi, utilizzarli nella pulizia.

Per eliminare le risorse del tutorial

1. Utilizzate il comando delete-deployment-group per eliminare il gruppo di distribuzione. CodeDeploy

```
aws deploy delete-deployment-group --application-name ecs-demo-deployment-group --
deployment-group-name ecs-demo-dg --region aws-region-id
```

2. Utilizzate il comando delete-application per eliminare l'applicazione. CodeDeploy

```
aws deploy delete-application --application-name ecs-demo-deployment-group --
region aws-region-id
```

3. Usa il comando delete-function per eliminare la funzione hook Lambda.

```
aws lambda delete-function --function-name AfterAllowTestTraffic
```

4. Usa il comando delete-log-group per eliminare il tuo gruppo di log. CloudWatch

```
aws logs delete-log-group --log-group-name /aws/lambda/AfterAllowTestTraffic
```

Tutorial: Implementa una funzione CodeDeploy Lambda aggiornata con e AWS Modello di applicazione serverless

AWS SAM è un framework open source per la creazione di applicazioni serverless. Trasforma ed espande la sintassi YAML in un AWS SAM modello in CloudFormation sintassi per creare applicazioni serverless, come una funzione Lambda. Per ulteriori informazioni, consulta What is the Serverless Application Model? AWS

In questo tutorial, utilizzi AWS SAM per creare una soluzione che esegua le seguenti operazioni:

- Crea la tua funzione Lambda.
- Crea CodeDeploy l'applicazione e il gruppo di distribuzione.
- Crea due funzioni Lambda che eseguono test di convalida della distribuzione durante CodeDeploy gli hook del ciclo di vita.
- Rileva quando la funzione Lambda viene aggiornata. L'aggiornamento della funzione Lambda attiva una distribuzione CodeDeploy che sposta in modo incrementale il traffico di produzione dalla versione originale della funzione Lambda alla versione aggiornata.

> **Note**
>
> Questo tutorial richiede la creazione di risorse che potrebbero comportare addebiti sull'account AWS . Questi includono possibili addebiti per CodeDeploy CloudWatch, Amazon e AWS Lambda. Per ulteriori informazioni, consulta CodeDeploy i prezzi, CloudWatch i prezzi di Amazon e AWS Lambda i prezzi.

Argomenti

- Prerequisiti

- Fase 1: Configura la tua infrastruttura
- Fase 2: Aggiornare la funzione Lambda
- Fase 3: Implementazione della funzione Lambda aggiornata
- Fase 4: Visualizza i risultati della distribuzione
- Fase 5: rimozione

Prerequisiti

Per completare questo tutorial, è necessario innanzitutto:

- Completa le fasi descritte in Guida introduttiva con CodeDeploy.
- Installa la AWS Serverless Application Model CLI. Per informazioni, consulta Installare la CLI AWS SAM.
- Crea un bucket S3. AWS SAM carica gli artefatti a cui si fa riferimento nel modello SAM in questo AWS bucket.

Fase 1: Configura la tua infrastruttura

Questo argomento mostra come utilizzare AWS SAM per creare file per il AWS SAM modello e le funzioni Lambda. Quindi, usi i `deploy` comandi AWS SAM `package` and per generare i componenti della tua infrastruttura. Quando l'infrastruttura è pronta, si dispone di un gruppo di CodeDeploy applicazioni e di distribuzione, della funzione Lambda per l'aggiornamento e la distribuzione e di due funzioni Lambda che contengono test di convalida eseguiti quando si distribuisce la funzione Lambda. Al termine, puoi utilizzarli CloudFormation per visualizzare i componenti nella console Lambda o AWS CLI per testare la funzione Lambda.

Argomenti

- Crea i tuoi file
- Package AWS Applicazione SAM
- Implementa il AWS Applicazione SAM
- (Facoltativo) ispeziona e testa la tua infrastruttura

Crea i tuoi file

Per creare l'infrastruttura, è necessario creare i file seguenti:

- `template.yml`
- `myDateTimeFunction.js`
- `beforeAllowTraffic.js`
- `afterAllowTraffic.js`

Argomenti

- Crea il tuo AWS Modello SAM
- Crea un file per la tua funzione Lambda
- Crea un file per la tua funzione BeforeAllowTraffic Lambda
- Crea un file per la tua funzione AfterAllowTraffic Lambda

Crea il tuo AWS Modello SAM

Crea un file modello AWS SAM che specifichi i componenti della tua infrastruttura.

Per creare il tuo AWS Modello SAM

1. Crea una directory denominata `SAM-Tutorial`.
2. Nella directory `SAM-Tutorial`, creare un file denominato `template.yml`.
3. Copiare il seguente codice YAML in `template.yml`. Questo è il modello AWS SAM .

```
AWSTemplateFormatVersion : '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: A sample SAM template for deploying Lambda functions.

Resources:
# Details about the myDateTimeFunction Lambda function
  myDateTimeFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: myDateTimeFunction.handler
      Runtime: nodejs18.x
# Instructs your myDateTimeFunction is published to an alias named "live".
      AutoPublishAlias: live
# Grants this function permission to call lambda:InvokeFunction
      Policies:
        - Version: "2012-10-17"
          Statement:
```

```
          - Effect: "Allow"
            Action:
              - "lambda:InvokeFunction"
            Resource: '*'
      DeploymentPreference:
# Specifies the deployment configuration
          Type: Linear10PercentEvery1Minute
# Specifies Lambda functions for deployment lifecycle hooks
          Hooks:
            PreTraffic: !Ref beforeAllowTraffic
            PostTraffic: !Ref afterAllowTraffic

# Specifies the BeforeAllowTraffic lifecycle hook Lambda function
  beforeAllowTraffic:
    Type: AWS::Serverless::Function
    Properties:
      Handler: beforeAllowTraffic.handler
      Policies:
        - Version: "2012-10-17"
# Grants this function permission to call
 codedeploy:PutLifecycleEventHookExecutionStatus
          Statement:
          - Effect: "Allow"
            Action:
              - "codedeploy:PutLifecycleEventHookExecutionStatus"
            Resource:
              !Sub 'arn:aws:codedeploy:${AWS::Region}:
${AWS::AccountId}:deploymentgroup:${ServerlessDeploymentApplication}/*'
        - Version: "2012-10-17"
# Grants this function permission to call lambda:InvokeFunction
          Statement:
          - Effect: "Allow"
            Action:
              - "lambda:InvokeFunction"
            Resource: !Ref myDateTimeFunction.Version
      Runtime: nodejs18.x
# Specifies the name of the Lambda hook function
      FunctionName: 'CodeDeployHook_beforeAllowTraffic'
      DeploymentPreference:
        Enabled: false
      Timeout: 5
      Environment:
        Variables:
          NewVersion: !Ref myDateTimeFunction.Version
```

```
# Specifies the AfterAllowTraffic lifecycle hook Lambda function
  afterAllowTraffic:
    Type: AWS::Serverless::Function
    Properties:
      Handler: afterAllowTraffic.handler
      Policies:
        - Version: "2012-10-17"
          Statement:
# Grants this function permission to call
 codedeploy:PutLifecycleEventHookExecutionStatus
          - Effect: "Allow"
            Action:
              - "codedeploy:PutLifecycleEventHookExecutionStatus"
            Resource:
              !Sub 'arn:aws:codedeploy:${AWS::Region}:
${AWS::AccountId}:deploymentgroup:${ServerlessDeploymentApplication}/*'
        - Version: "2012-10-17"
          Statement:
# Grants this function permission to call lambda:InvokeFunction
          - Effect: "Allow"
            Action:
              - "lambda:InvokeFunction"
            Resource: !Ref myDateTimeFunction.Version
      Runtime: nodejs18.x
# Specifies the name of the Lambda hook function
      FunctionName: 'CodeDeployHook_afterAllowTraffic'
      DeploymentPreference:
        Enabled: false
      Timeout: 5
      Environment:
        Variables:
          NewVersion: !Ref myDateTimeFunction.Version
```

Questo modello specifica quanto segue. Per ulteriori informazioni, consulta i concetti relativi ai AWS SAM modelli.

Una funzione Lambda chiamata **`myDateTimeFunction`**

Quando questa funzione Lambda viene pubblicata, la `AutoPublishAlias` riga del modello la collega a un alias denominato. `live` Più avanti in questo tutorial, un aggiornamento di questa

funzione attiva una distribuzione AWS CodeDeploy che sposta in modo incrementale il traffico di produzione dalla versione originale alla versione aggiornata.

Due funzioni di convalida dell'implementazione Lambda

Le seguenti funzioni Lambda vengono eseguite durante gli hook del ciclo di CodeDeploy vita. Le funzioni contengono codice che convalida la distribuzione del codice `myDateTimeFunction` aggiornato. Il risultato dei test di convalida viene passato a CodeDeploy utilizzando il relativo metodo API `PutLifecycleEventHookExecutionStatus`. Se un test di convalida ha esito negativo, la distribuzione ha esito negativo e viene eseguito il rollback.

- `CodeDeployHook_beforeAllowTraffic` viene eseguito durante l'hook `BeforeAllowTraffic`.
- `CodeDeployHook_afterAllowTraffic` viene eseguito durante l'hook `AfterAllowTraffic`.

Il nome di entrambe le funzioni inizia con `CodeDeployHook_`. Il `CodeDeployRoleForLambda` ruolo consente chiamate al `invoke` metodo Lambda solo nelle funzioni Lambda con nomi che iniziano con questo prefisso. Per ulteriori informazioni, consulta le pagine AppSpec sezione 'hooks' per un AWS Implementazione Lambda e PutLifecycleEventHookExecutionStatus nella Documentazione di riferimento dell'API CodeDeploy .

Rilevamento automatico di una funzione Lambda aggiornata

Il termine `AutoPublishAlias` indica al framework di rilevare quando la funzione `myDateTimeFunction` cambia e quindi distribuirla utilizzando l'alias `live`.

Una configurazione della distribuzione

La configurazione di distribuzione determina la velocità con cui l' CodeDeployapplicazione sposta il traffico dalla versione originale della funzione Lambda alla nuova versione. Questo modello specifica la configurazione della distribuzione predefinita `Linear10PercentEvery1Minute`.

> **Note**
>
> Non è possibile specificare una configurazione di distribuzione personalizzata in un modello AWS SAM. Per ulteriori informazioni, consulta Create a Deployment Configuration.

Funzioni hook del ciclo di vita della distribuzione

La sezione `Hooks` specifica le funzioni da eseguire durante gli hook degli eventi del ciclo di vita. `PreTraffic` specifica la funzione che viene eseguita durante l'hook `BeforeAllowTraffic`. `PostTraffic` specifica la funzione che viene eseguita durante l'hook `AfterAllowTraffic`.

Autorizzazioni per Lambda per richiamare un'altra funzione Lambda

L'`lambda:InvokeFunction`autorizzazione specificata concede al ruolo utilizzato dall'applicazione AWS SAM il permesso di richiamare una funzione Lambda. Ciò è necessario quando `CodeDeployHook_afterAllowTraffic` le funzioni `CodeDeployHook_beforeAllowTraffic` and richiamano la funzione Lambda distribuita durante i test di convalida.

Crea un file per la tua funzione Lambda

Creare il file per la funzione da aggiornare e distribuire più avanti in questo tutorial.

> **Note**
>
> Una funzione Lambda può utilizzare qualsiasi runtime supportato da. AWS Lambda Per ulteriori informazioni, consulta Tempi di esecuzione AWS Lambda.

Per creare la tua funzione Lambda

1. Creare un file di testo e salvarlo come `myDateTimeFunction.js` nella directory `SAM-Tutorial`.
2. Copia il seguente Node.js codice 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;
```

```
    }
};
```

La funzione Lambda restituisce il giorno, il mese e l'anno di ieri, oggi o domani. Più avanti in questo tutorial, si decomprimerà il codice che aggiorna la funzione per restituire informazioni sul giorno o sull'ora specificati (ad esempio, giorno, mese e anno o ora, minuto e secondo correnti). Il framework creato da AWS SAM rileva e distribuisce la versione aggiornata della funzione.

> **Note**
>
> Questa funzione Lambda viene utilizzata anche in un AWS Cloud9 tutorial. AWS Cloud9 è un ambiente di sviluppo integrato basato sul cloud. Per informazioni su come creare, eseguire, aggiornare ed eseguire il debug di questa funzione in AWS Cloud9, consulta AWS Lambda il tutorial per. AWS Cloud9

Crea un file per la tua funzione BeforeAllowTraffic Lambda

Crea il file per la tua funzione Lambda dell'`beforeAllowTraffic`hook.

1. Creare un file di testo e salvarlo come `beforeAllowTraffic.js` nella directory `SAM-Tutorial`.
2. Copia il seguente Node.js codice in. `beforeAllowTraffic.js` Questa funzione viene eseguita durante l'hook `BeforeAllowTraffic` della distribuzione.

```
'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");
       }
      });
     }
    });
   }
```

Crea un file per la tua funzione AfterAllowTraffic Lambda

Crea il file per la tua funzione Lambda dell'`afterAllowTraffic`hook.

1. Creare un file di testo e salvarlo come `afterAllowTraffic.js` nella directory `SAM-Tutorial`.
2. Copia il seguente Node.js codice in. `afterAllowTraffic.js` Questa funzione viene eseguita durante l'hook `AfterAllowTraffic` della distribuzione.

```
'use strict';

   const AWS = require('aws-sdk');
   const codedeploy = new AWS.CodeDeploy({apiVersion: '2014-10-06'});
   var lambda = new AWS.Lambda();

   exports.handler = (event, context, callback) => {

    console.log("Entering PostTraffic Hook!");

    // Read the DeploymentId and LifecycleEventHookExecutionId from the event
 payload
     var deploymentId = event.DeploymentId;
    var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;
```

```
var functionToTest = process.env.NewVersion;
console.log("AfterAllowTraffic hook tests started");
console.log("Testing new function version: " + functionToTest);

// Create parameters to pass to the updated Lambda function that
// include the original "date" parameter. If the function did not
// update as expected, then the "date" option might be invalid. If
// the parameter is invalid, the function returns
// a statusCode of 400 indicating it failed.
var lambdaParams = {
 FunctionName: functionToTest,
 Payload: "{\"option\": \"date\", \"period\": \"today\"}",
 InvocationType: "RequestResponse"
};

var lambdaResult = "Failed";
// Invoke the updated Lambda function.
lambda.invoke(lambdaParams, function(err, data) {
 if (err){ // an error occurred
  console.log(err, err.stack);
  lambdaResult = "Failed";
 }
 else{ // successful response
  var result = JSON.parse(data.Payload);
  console.log("Result: " +  JSON.stringify(result));
     console.log("statusCode: " + result.statusCode);

     // Check if the status code returned by the updated
     // function is 400. If it is, then it failed. If
     // is not, then it succeeded.
  if (result.statusCode != "400"){
       console.log("Validation of time parameter succeeded");
   lambdaResult = "Succeeded";
     }
     else {
       console.log("Validation failed");
     }

  // Complete the PostTraffic Hook by sending CodeDeploy the validation status
  var params = {
   deploymentId: deploymentId,
   lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
   status: lambdaResult // status can be 'Succeeded' or 'Failed'
  };
```

```
      // Pass CodeDeploy the prepared validation test results.
      codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data)
{
       if (err) {
        // Validation failed.
        console.log("CodeDeploy Status update failed");
        console.log(err, err.stack);
        callback("CodeDeploy Status update failed");
       } else {
        // Validation succeeded.
        console.log("CodeDeploy status updated successfully");
        callback(null, "CodeDeploy status updated successfully");
       }
      });
     }
    });
   }
```

Package AWS Applicazione SAM

Ora, la directory `SAM-Tutorial` dovrebbe contenere quattro file:

- `beforeAllowTraffic.js`
- `afterAllowTraffic.js`
- `myDateTimeFunction.js`
- `template.yml`

Ora sei pronto per usare il sam package comando AWS SAM per creare e impacchettare artefatti per le tue funzioni e applicazioni Lambda. CodeDeploy Gli artefatti vengono caricati in un bucket S3. L'output del comando è un nuovo file denominato `package.yml`. Questo file viene utilizzato dal sam deploy comando AWS SAM nel passaggio successivo.

> **Note**
>
> Per ulteriori informazioni sul sam package comando, consulta il riferimento ai comandi AWS SAM CLI nella AWS Serverless Application Model Developer Guide.

Nella directory `SAM-Tutorial`, eseguire il seguente comando.

```
sam package \
  --template-file template.yml \
  --output-template-file package.yml \
  --s3-bucket amzn-s3-demo-bucket
```

Per il `s3-bucket` parametro, specifica il bucket Amazon S3 che hai creato come prerequisito per questo tutorial. `output-template-file`specifica il nome del nuovo file utilizzato dal comando SAM. AWS sam deploy

Implementa il AWS Applicazione SAM

Usa il sam deploy comando AWS SAM con il `package.yml` file per creare le funzioni Lambda, CodeDeploy l'applicazione e il gruppo di distribuzione utilizzando. CloudFormation

> **Note**
>
> Per ulteriori informazioni sul sam deploy comando, consulta il riferimento ai comandi AWS SAM CLI nella AWS Serverless Application Model Developer Guide.

Nella directory `SAM-Tutorial`, eseguire il seguente comando.

```
sam deploy \
  --template-file package.yml \
  --stack-name my-date-time-app \
  --capabilities CAPABILITY_IAM
```

Il `--capabilities CAPABILITY_IAM` parametro è necessario per autorizzare la creazione CloudFormation di ruoli IAM.

(Facoltativo) ispeziona e testa la tua infrastruttura

Questo argomento mostra come visualizzare i componenti dell'infrastruttura e testare la funzione Lambda.

Per vedere il risultato del tuo stack dopo aver eseguito sam deploy

1. Apri la CloudFormation console all'indirizzo. https://console.aws.amazon.com/cloudformation

2. Nel riquadro di navigazione selezionare Stacks (Stack). Lo stack `my-date-time-app` viene visualizzato nella parte superiore.
3. Scegliere la scheda Events (Eventi) per visualizzare quali eventi sono completi. È possibile visualizzare gli eventi mentre è in corso la creazione dello stack. Al termine della creazione dello stack, è possibile visualizzare tutti gli eventi di creazione dello stack.
4. Con lo stack selezionato, scegliere Resources (Risorse). Nella colonna Tipo, puoi vedere le tue funzioni Lambda, `myDateTimeFunctionCodeDeployHook_beforeAllowTraffic`, e. `CodeDeployHook_afterAllowTraffic` La colonna Physical ID di ciascuna delle tue funzioni Lambda contiene un link per visualizzare le funzioni nella console Lambda.

   > **Note**
   >
   > Il nome della funzione `myDateTimeFunction` Lambda è preceduto dal nome dello CloudFormation stack e ha un identificatore aggiunto, quindi sembra. `my-date-time-app-myDateTimeFunction-123456ABCDEF`

5. Apri la console all'indirizzo. CodeDeploy https://console.aws.amazon.com/codedeploy/
6. Nel riquadro di navigazione espandere Deploy (Distribuisci) e scegliere Applications (Applicazioni).
7. Dovresti vedere una nuova CodeDeploy applicazione creata da CloudFormation con un nome che inizia con`my-date-time-app-ServerlessDeploymentApplication`. Scegliere questa applicazione.
8. Viene visualizzato un gruppo di distribuzione con un nome che inizia con `my-date-time-app-myDateTimeFunctionDeploymentGroup`. Scegliere questo gruppo di distribuzione.

   In Configurazione di distribuzione, dovresti vedere CodeDeployDefault.LambdaLinear10PercentEvery1Minute.

(Facoltativo) per testare la tua funzione (console)

1. Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Nel riquadro di navigazione, scegliere la funzione `my-date-time-app-myDateTimeFunction`. Nella console, il suo nome contiene un identificatore, quindi è simile a `my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Scegli Test (Esegui test).
4. In Event name (Nome evento), immettere un nome per l'evento di test.

5. Immettere quanto segue per l'evento di test, quindi scegliere Create (Crea).

```
{
  "option": "date",
  "period": "today"
}
```

6. Scegli Test (Esegui test). Viene visualizzato solo l'evento di test nell'elenco degli eventi di test.

   Per Execution result (Risultato esecuzione), viene visualizzato succeeded (riuscito).

7. In Execution result (Risultato esecuzione), espandere Details (Dettagli) per visualizzare i risultati. Vengono visualizzati il mese, il giorno e l'anno corrente.

(Facoltativo) per testare la funzionalità (AWS CLI)

1. Individua l'ARN della tua funzione Lambda. Viene visualizzato nella parte superiore della console Lambda quando visualizzi la funzione.
2. Eseguire il seguente comando seguente. Sostituire *`your-function-arn`* con la funzione ARN.

```
aws lambda invoke \
--function your-function-arn \
--cli-binary-format raw-in-base64-out \
--payload "{\"option\": \"date\", \"period\": \"today\"}" out.txt
```

3. Aprire `out.txt` per confermare che il risultato contenga il mese, il giorno e l'anno corrente.

Fase 2: Aggiornare la funzione Lambda

Questo argomento illustra come aggiornare il file `myDateTimeFunction.js`. Nella fase successiva, si utilizzerà il file per distribuire la funzione aggiornata. Ciò avvia la distribuzione CodeDeploy spostando il traffico di produzione dalla versione corrente della funzione Lambda alla versione aggiornata.

Per aggiornare la funzione Lambda

1. Aprire `myDateTimeFunction.js`.
2. Rimuovere i due contrassegni di commento ("`/*`" e "`*/`") e il testo esplicativo all'inizio e alla fine del `case` denominato `time` nel blocco `switch`.

Il codice non commentato consente di passare un nuovo parametro `time` alla funzione. Se si passa `time` alla funzione aggiornata, restituisce i `hour`, `minute` e `second` correnti.

3. Salva `myDateTimeFunction.js`. Avrà un aspetto simile al seguente:

```
'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;
  }
};
```

Fase 3: Implementazione della funzione Lambda aggiornata

In questo passaggio, usi il tuo update `myDateTimeFunction.js` per aggiornare e avviare la distribuzione della tua funzione Lambda. Puoi monitorare l'avanzamento della distribuzione nella console CodeDeploy o AWS Lambda .

La `AutoPublishAlias: live` riga del AWS SAM modello fa sì che l'infrastruttura rilevi gli aggiornamenti delle funzioni che utilizzano l'`live`alias. Un aggiornamento della funzione attiva una distribuzione CodeDeploy che sposta il traffico di produzione dalla versione originale della funzione alla versione aggiornata.

I sam deploy comandi sam package and vengono utilizzati per aggiornare e attivare la distribuzione della funzione Lambda. Questi comandi sono stati eseguiti in Package AWS Applicazione SAM e Implementa il AWS Applicazione SAM.

Per distribuire la funzione Lambda aggiornata

1. Nella directory `SAM-Tutorial`, eseguire il seguente comando.

   ```
   sam package \
     --template-file template.yml \
     --output-template-file package.yml  \
     --s3-bucket amzn-s3-demo-bucket
   ```

   Questo crea un nuovo set di artefatti che fanno riferimento alla funzione Lambda aggiornata nel bucket S3.
2. Nella directory `SAM-Tutorial`, eseguire il seguente comando.

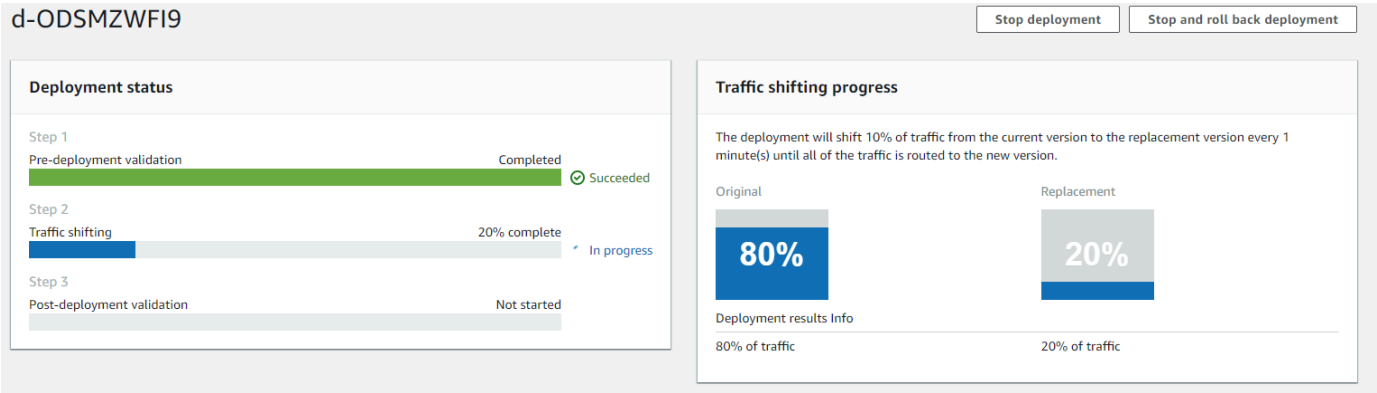
```
sam deploy \
  --template-file package.yml \
  --stack-name my-date-time-app \
  --capabilities CAPABILITY_IAM
```

Poiché il nome dello stack è fissomy-date-time-app, CloudFormation riconosce che si tratta di un aggiornamento dello stack. Per visualizzare lo stack aggiornato, torna alla CloudFormation console e, dal pannello di navigazione, scegli Stacks.

(Facoltativo) per visualizzare il traffico durante una distribuzione (console) CodeDeploy

1. Apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy/.
2. Nel riquadro di navigazione, espandi Applicazioni, quindi scegli la tua applicazione my-date-time-app -. ServerlessDeploymentApplication
3. In Deployment groups (Gruppi di distribuzione), scegliere il gruppo di distribuzione dell'applicazione. Il suo stato deve essere In progress (In corso).
4. In Deployment group history (Cronologia gruppo di distribuzione), scegliere la distribuzione in corso.

   La barra di avanzamento di Traffic shifting (Trasferimento traffico) e le percentuali nelle caselle Original (Origine) e Replacement (Sostituzione) di questa pagina visualizzano l'avanzamento.

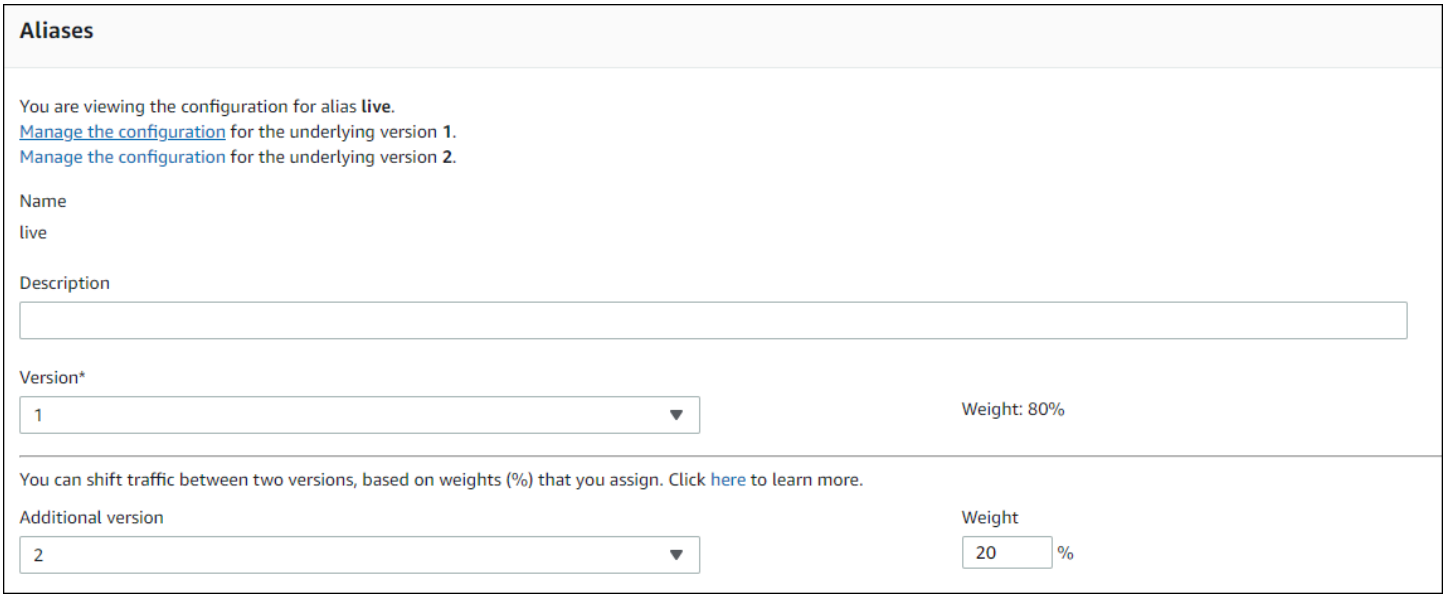


(Facoltativo) per visualizzare il traffico durante una distribuzione (console Lambda)

1. Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.

2. Nel riquadro di navigazione, scegliere la funzione `my-date-time-app-myDateTimeFunction`. Nella console, il suo nome contiene un identificatore, quindi è simile a `my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Scegli Alias, quindi scegli live.

I pesi accanto alla versione originale della funzione (versione 1) e alla versione della funzione aggiornata (versione 2) mostrano quanto traffico viene servito a ciascuna versione nel momento in cui questa pagina della AWS Lambda console è stata caricata. La pagina non aggiorna i pesi nel tempo. Se aggiorni la pagina una volta al minuto, il peso per la versione 1 diminuisce del 10% e il peso per la versione 2 aumenta del 10% fino a quando il peso per la versione 2 è 100.



Fase 4: Visualizza i risultati della distribuzione

In questa fase, vengono visualizzati i risultati della distribuzione. Se l'implementazione ha esito positivo, puoi confermare che la funzione Lambda aggiornata riceva traffico di produzione. Se la distribuzione fallisce, puoi utilizzare CloudWatch Logs per visualizzare l'output dei test di convalida nella funzione Lambda che viene eseguita durante gli hook del ciclo di vita della distribuzione.

Argomenti

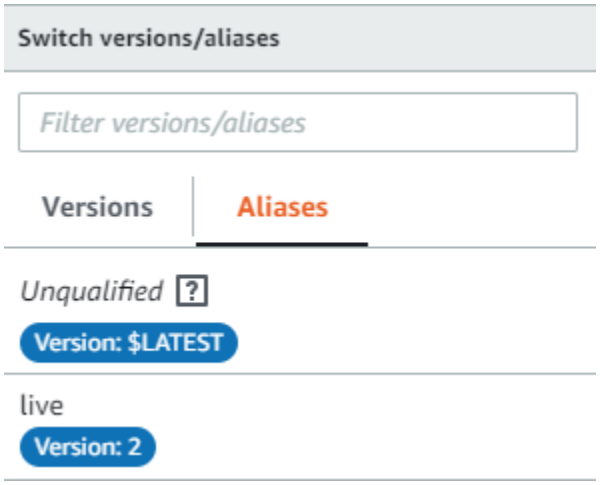
- Metti alla prova la funzione distribuita
- Visualizza gli eventi degli hook nei registri CloudWatch

Metti alla prova la funzione distribuita

Il sam deploy comando aggiorna la funzione `my-date-time-app-myDateTimeFunction` Lambda. La versione della funzione viene aggiornata a 2 e aggiunta all'alias `live`.

Per vedere l'aggiornamento nella console Lambda

1. Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Nel riquadro di navigazione, scegliere la funzione `my-date-time-app-myDateTimeFunction`. Nella console, il suo nome contiene un identificatore, quindi è simile a `my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Scegliere Qualifiers (Qualificatori), quindi Aliases (Alias). Una volta completata la distribuzione (circa 10 minuti), per l'alias `live`, viene visualizzato Version: 2 (Versione: 2).



4. In Function code (Codice funzione), visualizzare il codice sorgente per la funzione. Verrà visualizzata la modifica.
5. (Facoltativo) È possibile utilizzare le istruzioni di test in Fase 2: Aggiornare la funzione Lambda per testare la funzione aggiornata. Creare un nuovo evento di test con il payload seguente, quindi confermare che il risultato contenga l'ora, il minuto e il secondo correnti.

   ```
   {
       "option": "time"
     }
   ```

   Per utilizzare la funzione AWS CLI per testare la funzione aggiornata, esegui il comando seguente, quindi apri `out.txt` per confermare che il risultato contenga l'ora, il minuto e il secondo correnti.

```
aws lambda invoke --function your-function-arn --payload "{\"option\": \"time\"}"
 out.txt
```

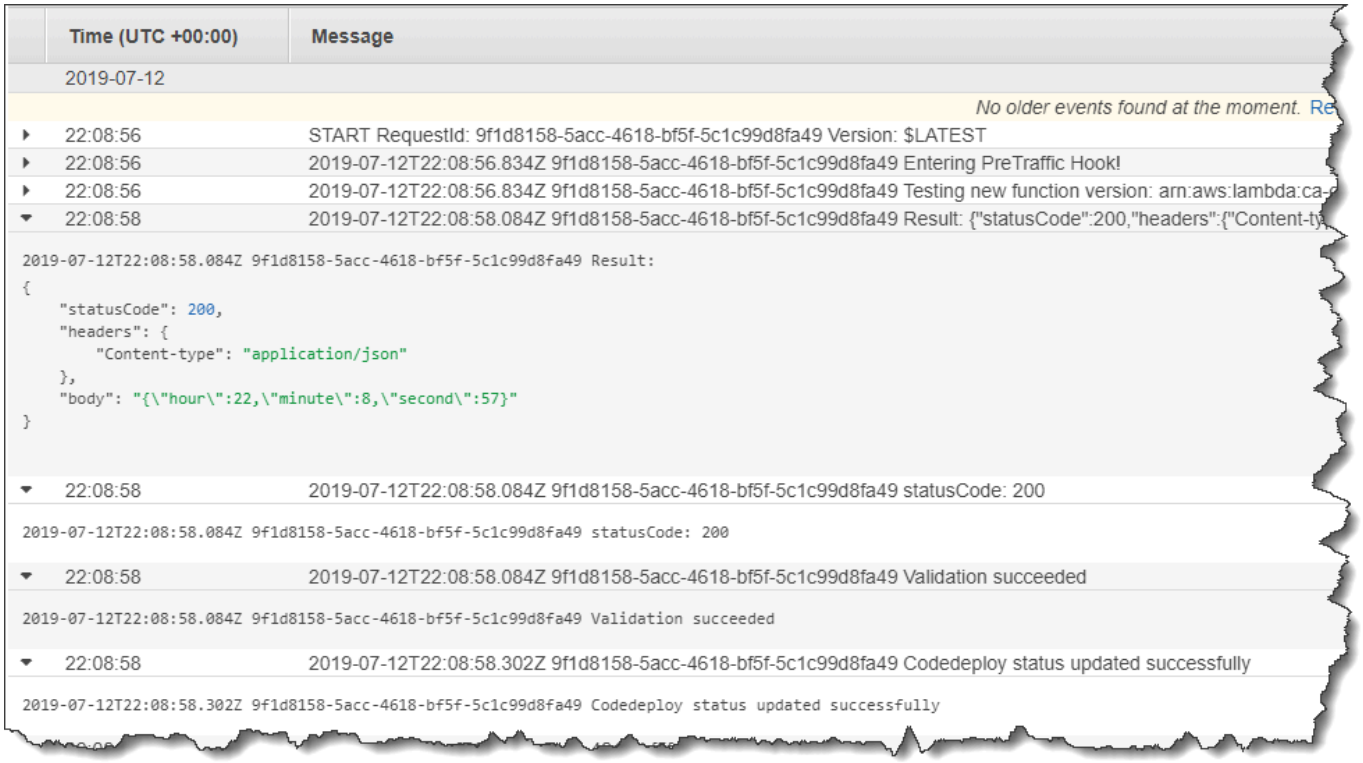
> **Note**
>
> Se utilizzi il AWS CLI per testare la tua funzione prima del completamento della distribuzione, potresti ricevere risultati imprevisti. Questo perché ogni minuto sposta CodeDeploy in modo incrementale il 10 percento del traffico verso la versione aggiornata. Durante la distribuzione, il traffico continua a puntare alla versione originale, quindi `aws lambda invoke` potrebbe utilizzare la versione originale. Dopo 10 minuti, la distribuzione è completa e tutto il traffico punta alla nuova versione della funzione.

Visualizza gli eventi degli hook nei registri CloudWatch

Durante l'`BeforeAllowTraffic`hook, CodeDeploy esegue la funzione `CodeDeployHook_beforeAllowTraffic` Lambda. Durante l'`AfterAllowTraffic`hook, CodeDeploy esegue la funzione `CodeDeployHook_afterAllowTraffic` Lambda. Ogni funzione esegue un test di convalida che richiama la versione aggiornata della funzione utilizzando il nuovo parametro `time`. Se l'aggiornamento della funzione Lambda ha esito positivo, l'`time`opzione non causa errori e la convalida ha esito positivo. Se la funzione non è stata aggiornata, il parametro non riconosciuto genera un errore e la convalida ha esito negativo. Questi test di convalida sono solo a scopo dimostrativo. È possibile scrivere i test per convalidare la distribuzione. Puoi utilizzare la console CloudWatch Logs per visualizzare i test di convalida.

Per visualizzare i tuoi eventi hook CodeDeploy

1. Apri la CloudWatch console su https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione scegliere Log.
3. Dall'elenco dei gruppi di log, scegli/aws/lambda/CodeDeployHook_before AllowTraffic o/aws/lambda/CodeDeployHookAllowTraffic_after.
4. Scegli il flusso di log. Ne viene visualizzato solo uno.
5. Espandere gli eventi per visualizzarne i dettagli.



Fase 5: rimozione

Per evitare ulteriori addebiti per le risorse utilizzate durante questo tutorial, elimina le risorse create dal AWS SAM modello e CloudWatch i log creati dalle funzioni di convalida Lambda.

Per eliminare i tuoi CloudFormation stack

1. Accedi a Console di gestione AWS e apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Nella colonna Stacks (Stack) scegliere lo stack `my-date-time-app`, quindi Delete (Elimina).
3. Quando viene richiesto, scegliere nuovamente Delete stack (Elimina stack). Le funzioni Lambda, il gruppo di CodeDeploy applicazioni e distribuzione e i ruoli IAM creati da AWS SAM vengono eliminati.

Per eliminare i log in Logs CloudWatch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel riquadro di navigazione scegliere Log.

3. Dall'elenco dei gruppi di log, scegli il pulsante accanto a/aws/lambda/CodeDeployHook_before AllowTraffic.
4. In Actions (Operazioni), scegliere Delete log group (Elimina gruppo di log), quindi Yes, Delete (Sì, elimina).
5. Dall'elenco dei gruppi di log, scegliete il pulsante accanto a/aws/lambda/ CodeDeployHookAllowTraffic_after.
6. In Actions (Operazioni), scegliere Delete log group (Elimina gruppo di log), quindi Yes, Delete (Sì, elimina).

Collaborazione con l' CodeDeploy agente

L' AWS CodeDeploy agente è un pacchetto software che, una volta installato e configurato su un'istanza, ne consente l'utilizzo nelle CodeDeploy distribuzioni.

AWS supporta attivamente l'ultima versione secondaria dell' CodeDeploy agente. Per l'ultima versione rilasciata, vedereCronologia delle versioni dell'agente CodeDeploy.

> **Note**
>
> L' CodeDeploy agente è necessario solo se si esegue la distribuzione su una piattaforma di EC2/On-Premises elaborazione. L'agente non è necessario per le distribuzioni che utilizzano la piattaforma di calcolo Amazon ECS o Lambda AWS .

Un file di configurazione viene posizionato sull'istanza quando viene installato l'agente. Questo file viene utilizzato per specificare il modo in cui viene utilizzato l'agente. Questo file di configurazione specifica i percorsi delle directory e altre impostazioni da utilizzare durante l'interazione AWS CodeDeploy con l'istanza. Puoi modificare alcune delle opzioni di configurazione nel file. Per informazioni sull'utilizzo del file di configurazione dell' CodeDeploy agente, vedere. CodeDeploy riferimento alla configurazione dell'agente

Per ulteriori informazioni sull'utilizzo dell' CodeDeploy agente, ad esempio sui passaggi per l'installazione, l'aggiornamento e la verifica delle versioni, vedereGestione delle operazioni degli CodeDeploy agenti.

Argomenti

- Sistemi operativi supportati dall'agente CodeDeploy
- Protocollo di comunicazione e porta per l'agente CodeDeploy
- Cronologia delle versioni dell'agente CodeDeploy
- Gestione del processo CodeDeploy
- Revisione dell'applicazione e pulizia dei file di registro
- File installati dall'agente CodeDeploy
- Gestione delle operazioni degli CodeDeploy agenti

Sistemi operativi supportati dall'agente CodeDeploy

Sistemi operativi AMI Amazon EC2 supportati

L' CodeDeploy agente è stato testato sui seguenti sistemi operativi AMI Amazon EC2:

- Amazon Linux 2023 (x86_64, aarch64)
- Amazon Linux 2 (x86_64, aarch64)
- Red Hat Enterprise Linux (RHEL) 8, 9, 10 (x86_64, aarch64)
- Oracle Linux 8, 9, 10 (x86_64, aarch64)
- Rocky Linux 9, 10 (x86_64, aarch64)
- CentOS Stream 9, 10 (x86_64, aarch64)
- SALES 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 su istanze Mac EC2 con Apple silicon (aarch64)

> **Note**
>
> Le seguenti piattaforme sono state testate solo sulla versione CodeDeploy agente 2.0.x e successive: Windows Server 2025, RHEL 10, Oracle Linux 8, 9, 10, Rocky Linux 9, 10, CentOS Stream 9, 10, SLES 15, Debian 11, 12, 13 e Ubuntu Server 24.04, 25.04, 26.04. macOS è supportato solo nella versione dell' CodeDeploy agente 2.1.2 e successive. L'agente supporta macOS 15, 26, 27 con Apple silicon.

L' CodeDeploy agente è disponibile come open source per consentirti di adattarlo alle tue esigenze. Può essere utilizzato con altri sistemi operativi AMI Amazon EC2. Per ulteriori informazioni, vai al repository degli CodeDeploy agenti in. GitHub

Sistemi operativi locali supportati

L' CodeDeploy agente è stato testato sui seguenti sistemi operativi locali:

- Windows Server 2016, 2019, 2022, 2025 (x86_64)
- Red Hat Enterprise Linux (RHEL) 8, 9, 10 (x86_64, aarch64)
- Oracle Linux 8, 9, 10 (x86_64, aarch64)
- Rocky Linux 9, 10 (x86_64, aarch64)
- CentOS Stream 9, 10 (x86_64, aarch64)
- SALES 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 con Apple silicon (aarch64)

> **Note**
>
> Le seguenti piattaforme sono state testate solo sulla versione CodeDeploy agente 2.0.x e successive: Windows Server 2025, RHEL 10, Oracle Linux 8, 9, 10, Rocky Linux 9, 10, CentOS Stream 9, 10, SLES 15, Debian 11, 12, 13 e Ubuntu Server 24.04, 25.04, 26.04. macOS è supportato solo nella versione dell' CodeDeploy agente 2.1.2 e successive. L'agente supporta macOS 15, 26, 27 con Apple silicon.

L' CodeDeploy agente è disponibile come open source per consentirti di adattarlo alle tue esigenze. Può essere utilizzato con altri sistemi operativi su istanze locali. Per ulteriori informazioni, vai al repository degli CodeDeploy agenti in GitHub.

Protocollo di comunicazione e porta per l'agente CodeDeploy

L' CodeDeploy agente comunica in uscita utilizzando HTTPS sulla porta 443.

Quando l' CodeDeploy agente viene eseguito su un'istanza EC2, utilizzerà l'endpoint di metadati EC2 per recuperare le informazioni relative all'istanza. Ulteriori informazioni su come limitare e concedere l'accesso al servizio di metadati delle istanze.

Cronologia delle versioni dell'agente CodeDeploy

Le tue istanze devono eseguire una versione supportata dell' CodeDeploy agente. La versione minima attualmente supportata è la 2.1.x.

> **Note**
>
> Si consiglia di utilizzare la versione più recente dell' CodeDeploy agente. Se riscontri problemi, esegui l'aggiornamento alla versione più recente prima di contattare l' AWS assistenza. Per informazioni sull'aggiornamento, consultaAggiorna l' CodeDeploy agente.

> **Stato di implementazione della versione 2.x**
>
> La versione 2.x dell' CodeDeploy agente è in fase di implementazione Regioni AWS e potrebbe non essere ancora disponibile in tutte le regioni. La versione 2.x è un aggiornamento opzionale. Per effettuare l'aggiornamento alla versione 2.x da qualsiasi versione precedente, vedere. Installa l' CodeDeploy agente Per scoprire quale versione è possibile installare, consulta. Determina l'ultima versione dell'agente disponibile in un Regione AWS

La tabella seguente elenca tutte le versioni dell' CodeDeploy agente e le funzionalità e i miglioramenti inclusi in ciascuna versione.

Versione	Data di rilascio	Informazioni
2.1.2	13 ottobre 2026	Aggiunto: Supporto per macOS 15, 26, 27 con Apple silicon. Per istruzioni, consulta l'argomento Installa l' CodeDeploy agente per macOS. Aggiunto: download a intervallo parallelo delle revisioni delle applicazioni da Amazon S3, disattivati per impostazione predefinita. Per ulteriori informazioni, consulta CodeDeploy riferimento alla configurazione dell'agente.

Versione	Data di rilascio	Informazioni
		Risolto: l'output dello script degli eventi del ciclo di vita viene scritto `codedeploy-agent-deployments.log` riga per riga, con il prefisso dell'ID di distribuzione.
2.1.1	2 ottobre 2026	Risolto: l'agente ora si connette all' CodeDeploy endpoint corretto nello AWS European Sovereign Cloud.
2.1.0	7 settembre 2026	Aggiunto: supporto nativo per la modalità di `RESTART` distribuzione, che riavvia l'applicazione sulle istanze di destinazione utilizzando la revisione dell'ultima distribuzione riuscita del gruppo di distribuzione. L'agente si installa dalla copia della revisione già presente sull'istanza. Se tale copia è mancante o incompleta, l'agente scarica la revisione, quindi la distribuzione viene comunque completata. Per la modalità di distribuzione stessa, consulta DeploymentMode nel riferimento API. AWS CodeDeploy Modificato (sicurezza): l'agente ora rifiuta una distribuzione il cui percorso AppSpec del file si risolve all'esterno della directory di revisione dell'applicazione. I percorsi annidati all'interno della revisione, ad esempio, sono ancora supportati. `configs/appspec.yml`
2.0.1	24 agosto 2026	Risolto: in Windows, i percorsi dei AppSpec file che iniziano con una barra rovesciata (\) o con più di un separatore (ad esempio,`//my-folder`) vengono ora risolti all'interno della revisione, non nella radice dell'unità. Per esempi di valori supportati`source`, vedere. AppSpec sezione 'file' (solo distribuzioni) EC2/On-Premises

Versione	Data di rilascio	Informazioni
2.0.0	15 luglio 2026	Modificato: l' CodeDeploy agente è ora disponibile Rust-based e viene fornito come un unico binario nativo autonomo. La dipendenza dal runtime di Ruby è stata eliminata. Ruby non è più necessario per installare o eseguire l'agente. Per impostazione predefinita, l'agente è compatibile dal punto di vista del comportamento con la versione 1.8.x; i file di configurazione esistenti, la AppSpec gestione, la semantica del lifecycle-hook, il layout su disco e i risultati della distribuzione rimangono invariati, a meno che non sia indicato di seguito. Aggiunto: supporto esteso per i sistemi operativi, tra cui Windows Server 2025, RHEL 10 e Ubuntu Server 26.04, oltre a distribuzioni Linux e varianti ARM (aarch64) aggiuntive. Per l'elenco completo delle piattaforme testate con questa versione, consulta. Sistemi operativi supportati dall'agente CodeDeploy Aggiunto: un `codedeploy-agent update` comando per l'aggiornamento automatico degli agenti su richiesta. Aggiunta: una porta di comando locale. Si tratta di un'interfaccia di gestione TCP opzionale che ascolta solo l'indirizzo di loopback (127.0.0.1) per interrogare lo stato dell'agente e iniettare i comandi localmente. È disabilitata per impostazione predefinita ed è abilitata con. `enable_command_port` Modificato: il comando di distribuzione locale è ora disponibile `codedeploy-agent deploy-local` . Il `codedeploy-local` comando rimane disponibile per la compatibilità con le versioni precedenti. Tutte le opzioni di comando vengono preservate e i percorsi locali, Amazon S3 (`s3://`) e GitHub le sorgenti in bundle sono ancora supportati. Modificato: su Linux, l'agente è gestito da un'unità systemd nativa (`codedeploy-agent.service`); lo script

Versione	Data di rilascio	Informazioni
		SysV init.d non è più installato. Utilizzare per gestire l'agente`systemctl` . Modificato (sicurezza): la verifica del certificato TLS è sempre abilitata e i core dump sono disabilitati per impostazione predefinita. Aggiunto: impostazioni di rafforzamento opzionali che puoi abilitare per soddisfare i tuoi requisiti di sicurezza. Ogni impostazione è disattivata per impostazione predefinita e puoi abilitare quelle desiderate singolarmente. È possibile rifiutare i pacchetti che contengono collegamenti simbolici, percorsi di attraversamento di percorsi o autorizzazioni non sicure, limitare l'ambiente degli hook di eventi del ciclo di vita e limitare le autorizzazioni delle directory degli agenti e dei file di registro. Per l'elenco completo, consulta CodeDeploy riferimento alla configurazione dell'agente. Modificato: lo script di installazione da riga di comando è ora scritto in Bash (precedentemente Ruby). Richiede una delle due opzioni. `curl` `wget` Modificato: lo script di installazione per la versione 2.0.x viene pubblicato con il `latestv2/` prefisso nei bucket regionali . `aws-codedeploy-` *`region-identifier`* Il `latest/` prefisso continua a servire lo script di installazione della versione 1.8.x. Modificato: AWS ora pubblica la versione 2.0.x come pacchetto `AWSCodeDeployAgentV2` Systems Manager Distributor. Il `AWSCodeDeployAgent` pacchetto continua a funzionare con la versione 1.8.x. Rimosso: l'impostazione `ssl_verify_peer` di configura zione e le chiavi di configurazione di Ruby ProcessManager .

Versione	Data di rilascio	Informazioni
		Importante: attualmente non esiste alcun percorso di aggiornamento automatico dalla versione 1.8.x alla 2.0.0. Per eseguire l'aggiornamento alla versione 2.0.0, installalo con il pacchetto `AWSCodeDeployAgentV2` Systems Manager Distributor o esegui lo script regionale `latestv2/install` manualmente. Per istruzioni, consulta l'argomento Installa l' CodeDeploy agente.
1.8.1	3 febbraio 2026	Risolto: bug corretto dell'endpoint S3. Aggiunto: certificato CA Sectigo all'elenco delle CA di Windows.
1.8.0	31 luglio 2025	Modificato: è stato aggiornato il pacchetto Ruby alla versione 3.2 nell' CodeDeploy agente per Windows.
1.7.1	14 novembre 2024	Modificato: dipendenze aggiornate per le patch di sicurezza.

Versione	Data di rilascio	Informazioni
1.7.0	6 marzo 2024	Aggiunto: un'impostazione `:disable_imds_v1:` di configurazione nel file di configurazione dell' CodeDeploy agente. Usa questa impostazione per disabilitare il fallback su IMDSv1 quando si verificano errori IMDSv2. L'impostazione predefinita è (abilita il fallback). `false` Per ulteriori informazioni, vedere il riferimento alla configurazione CodeDeploy dell'agente. Aggiunto: Supporto per il sistema operativo Red Hat Enterprise Linux 9 (RHEL 9). Aggiunto: Supporto per le versioni 3.1 e 3.2 di Ruby su Ubuntu Server. Risolto: l' CodeDeploy agente ora genera un errore intuitivo se il file di configurazione dell' CodeDeploy agente non viene caricato. Modificato: Ruby aggiornato alla versione 2.7.8-1 nell'agente per Windows. CodeDeploy
1.6.0	30 marzo 2023	Aggiunto: Supporto per Ruby 3.1, 3.2. Aggiunto: Supporto per Amazon Linux 2023. Aggiunto: Supporto per Windows Server 2022. Modificato: l'impostazione predefinita di `verbose` è ora `false` per le istanze di Windows Server. Per continuare a stampare i messaggi di debug nei file di registro su Windows, è necessario impostare `verbose` su. `true` Rimosso: Supporto per Windows Server 2016 e Windows Server 2012 R2. Rimosso: Supporto per Amazon Linux 2018.03.x.

Versione	Data di rilascio	Informazioni
1.5.0	3 marzo 2023	Aggiunto: Supporto per Ruby 3. Aggiunto: Supporto per Ubuntu 22.04. Risolto: un problema a causa del quale il riavvio dell' CodeDeploy agente subito dopo l'avvio causava il blocco dell'agente. Modificato: l' CodeDeploy agente ora fallisce la distribuzione dell'host all'avvio dell'agente se il servizio dell'agente si riavvia in modo imprevisto durante l'esecuzione di uno script hook. Questa correzione consente di evitare di attendere il periodo di timeout di 70 minuti prima di riprovare una distribuzione. Avviso di obsolescenza: l' CodeDeploy agente 1.5.0 è l'ultima versione a supportare Windows Server 2016 e Windows Server 2012 R2. Rimosso: Supporto per l' CodeDeployagente su Ubuntu 14.04 LTS, Windows Server 2008 R2 e Windows Server 2008 R2 a 32 bit.
1.4.1	6 dicembre 2022	Risolto: vulnerabilità di sicurezza relativa alla registrazione. Miglioramento: registrazione migliorata durante il polling del comando host.

Versione	Data di rilascio	Informazioni
1.4.0	31 agosto 2022	Aggiunto: Supporto per Red Hat Enterprise Linux 8. Aggiunto: supporto per percorsi di file lunghi sull' CodeDeploy agente per Windows. Per abilitare percorsi di file lunghi, è necessario impostare la chiave di registro di Windows appropriata e quindi riavviare l'agente. Per ulteriori informazioni, consulta I percorsi di file lunghi causano errori del tipo «Nessun file o directory di questo tipo». Risolto: un problema con l'operazione di decompressione quando il disco era pieno. L' CodeDeploy agente ora rileva il codice di uscita 50 dell'unzip che indica un disco completo, rimuove i file parzialmente estratti e solleva un'eccezione per segnalare un errore al server. CodeDeploy Il messaggio di errore è visibile come messaggio di errore relativo a un evento del ciclo di vita e la distribuzione a livello di host si interromperà senza essere bloccata o scaduta. Risolto: un problema che causava il fallimento dell'agente. Risolto: un problema che causava il timeout degli hook durante una gara irregolare. Gli hook senza script ora continueranno e non causeranno più guasti o timeout. Modificato: lo `update` script dalla `bin` directory dell' CodeDeploy agente è stato rimosso perché non viene più utilizzato. Modificato: l' CodeDeploy agente per Windows Server ora include Ruby 2.7. Modificato: sono state aggiunte nuove variabili di ambiente, che possono essere utilizzate dagli script hook a seconda dell'origine del pacchetto di distribuzione (Amazon GitHub S3 o).

Versione	Data di rilascio	Informazioni
		Per ulteriori informazioni, consulta Disponibilità delle variabili di ambiente per gli hook. **Important** Avviso di obsolescenza: CodeDeploy agent 1.4.0 è l'ultima versione che includerà i programmi di installazione per Windows Server a 32 bit. Avviso di obsolescenza: l' CodeDeployagente 1.4.0 è l'ultima versione che supporterà Windows Server 2008 R2. Rimosso: supporto per l' CodeDeploy agente sulle seguenti AMI Amazon EC2: Amazon Linux 2014.09, 2016.03, 2016.09 e 2017.03.

Versione	Data di rilascio	Informazioni
1.3.2	06 maggio 2021	**Important** CodeDeploy indirizzi dell'agente 1.3.2 che influiscono sugli host Windows che eseguono l'agente. CVE-2018-1000201 Il CVE cita ruby-ffi, che è una dipendenza dell'agente. CodeDeploy Se il tuo agente è stato installato con Amazon EC2 Systems Manager (SSM) ed è impostato per l'aggiornamento automatico, non è richiesta alcuna azione. Altrimenti, è necessaria un'azione per aggiornare manualmente l'agente. Per aggiornare l'agente, segui le istruzioni riportate in Aggiornamento dell' CodeDeploy agente su Windows Server. Risolto: un problema durante l'installazione dell' CodeDeploy agente su Ubuntu 20.04 e versioni successive. Risolto: un problema intermittente che si verificava durante l'estrazione di file compressi perché i percorsi relativi non venivano gestiti correttamente. Aggiunto: supporto AWS PrivateLink e endpoint VPC per istanze Windows. Aggiunti: miglioramenti ai AppSpec file, come descritto di seguito. • È ora possibile specificare un nome file personalizzato per il AppSpec file durante la creazione di una distribuzione locale. Per ulteriori informazioni, consulta Crea una distribuzione locale. • Il AppSpec file può ora avere un'`.yaml`estensione. • È ora possibile sovrascrivere i file distribuiti utilizzando una nuova `file_exists_behavior` impostazione

Versione	Data di rilascio	Informazioni
		opzionale nel AppSpec file. Per ulteriori informazioni, consulta AppSpec sezione 'file' (solo distribuzioni) EC2/On-Premises. Aggiornato: CodeDeploy ora utilizza l' AWS SDK per Ruby 3.0.
1.3.1	22 dicembre 2020	Risolto: problema 1.3.0 che impediva l'avvio delle istanze locali.
1.3.0	10 novembre 2020	**Important** Questa versione è obsoleta. Risolto: rimosso un certificato scaduto che non veniva più utilizzato. Risolto: è stato rimosso il messaggio di richiesta dallo script di disinstallazione dell'agente utilizzato da AWS Systems Manager, semplificando il downgrade di un host o di un parco macchine a una versione precedente dell'agente.
1.2.1	23 settembre 2020	Modificato: AWS SDK per Ruby dipendenza aggiornata dalla v2 alla v3. Aggiunto: Support per IMDSv2. Include un fallback silenzioso su IMDSv1 se le richieste http IMDSv2 falliscono. Modificato: sono state aggiornate le dipendenze Rake e Rubyzip per le patch di sicurezza. Risolto: assicurati che un file PID vuoto restituisca uno stato di `No CodeDeploy Agent Running` e pulisca il file PID all'avvio dell'agente.

Versione	Data di rilascio	Informazioni
1.1.2	4 agosto 2020	Aggiunto: Supporto per Ubuntu Server 19.10 e 20.04. Nota: la versione 19.10 ha raggiunto la data di fine del ciclo di vita e non è più supportata da Ubuntu o. CodeDeploy Aggiunto: miglioramenti dell'efficienza della memoria per Linux e Ubuntu per rilasciare la memoria riservata più tempestiv amente. Aggiunto: compatibilità con Windows Server «silent-cleanup», che in alcuni casi causava la mancata risposta dell'agente. Aggiunto: ignora le directory non vuote durante la pulizia per evitare errori durante la distribuzione. Aggiunto: Supporto per AWS Local Zone a Los Angeles (LA). Aggiunto: estrae AZ dai metadati dell'istanza per garantire la compatibilità per AWS Local Zones. Aggiunto: gli utenti possono ora fornire il proprio archivio nelle sottodirectory e non sono tenuti a memorizzarlo nella directory principale. Aggiunto: rilevato un problema con Rubyzip che potrebbe causare perdite di memoria. È stato aggiornato il comando unzip per tentare innanzitutto di utilizzare un'utilità di decompressione installata dal sistema prima di utilizzare Rubyzip. Aggiunto: `:enable_auth_policy:` come impostazione di configurazione dell'agente. Modificato: gli avvisi di unzip ora vengono ignorati, quindi le distribuzioni continueranno.

Versione	Data di rilascio	Informazioni
1.1.0	30 giugno 2020	Modificato: il controllo delle versioni dell' CodeDeployagente ora segue la convenzione di controllo delle versioni standard di Ruby. Aggiunta: aggiunto un nuovo parametro per il comando di installazione e aggiornamento per consentire l'installazione di una versione specifica dell'agente dalla riga di comando. Rimosso: rimosso l' CodeDeploy agente Auto Updater per Linux e Ubuntu. Per configurare gli aggiornamenti automatici dell' CodeDeploy agente, vedi Installare l' CodeDeploy agente utilizzando AWS Systems Manager.
1.0.1.1597	15 novembre 2018	Miglioramento: CodeDeploy supporta Ubuntu 18.04. Miglioramento: supporta Ruby 2.5. CodeDeploy Miglioramento: supporta gli endpoint FIPS. CodeDeploy Per ulteriori informazioni sugli endpoint FIPS, consulta la panoramica su FIPS 140-2. Per gli endpoint che possono essere utilizzati con CodeBuild, consulta Regioni ed endpoint. CodeDeploy
1.0.1.1518	12 giugno 2018	Miglioramento: è stato risolto un problema che causava un errore quando l' CodeDeploy agente veniva chiuso mentre accettava le richieste di sondaggio. Miglioramento: è stata aggiunta una funzionalità di tracciamento della distribuzione che impedisce la chiusura dell' CodeDeploy agente quando è in corso una distribuzione. Miglioramento: prestazioni superiori durante l'eliminazione di file.

Versione	Data di rilascio	Informazioni
1.0.1.1458	6 marzo 2018	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Miglioramento: le convalide dei certificati ora supportano più autorità attendibili. Miglioramento: è stato risolto un problema che causava il fallimento della CLI locale durante una distribuzione che includeva un evento del ciclo di vita. BeforeInstall Miglioramento: è stato risolto un problema che poteva causare il fallimento di una distribuzione attiva quando l'agente veniva aggiornato. CodeDeploy
1.0.1.1352	16 novembre 2017	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Funzionalità: è stata introdotta una nuova funzionalità per il test e il debug di una EC2/On-Premises distribuzione su un computer o un'istanza locale in cui è installato l'agente. CodeDeploy
1.0.1.1106	16 maggio 2017	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Funzionalità: introdotto il supporto per la gestione di contenuti in una posizione di destinazione che non era parte della revisione dell'applicazione della distribuzione riuscita più recente. Le opzioni delle distribuzioni per i contenuti esistenti ora includono la conservazione dei contenuti, la sovrascrittura dei contenuti o l'esito negativo della distribuzione. Miglioramento: reso l' CodeDeploy agente compatibile con la versione 2.9.2 di (aws-sdk-core 2.9.2). AWS SDK per Ruby

Versione	Data di rilascio	Informazioni
1.0.1.1095	29 marzo 2017	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Miglioramento: è stato introdotto il supporto per l' CodeDeploy agente nella regione Cina (Pechino). Miglioramento: ha abilitato l'esecuzione di Puppet su istanze di Windows Server quando richiamato da un hook di eventi del ciclo di vita. Miglioramento: migliorata la gestione delle operazioni `untar`.
1.0.1.1067	6 gennaio 2017	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Miglioramento: rivisti numerosi messaggi di errore per includere cause più specifiche per le distribuzioni con esito negativo. Miglioramento: è stato risolto un problema che impediva all' CodeDeploy agente di identificare la revisione corretta dell'applicazione da distribuire durante alcune distribuzioni. Miglioramento: ripristinato l'utilizzo di `pushd` e di `popd` prima e dopo l'operazione `untar`.
1.0.1.1045	21 novembre 2016	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Miglioramento: reso l' CodeDeploy agente compatibile con la versione 2.6.11 di (aws-sdk-core 2.6.11). AWS SDK per Ruby

Versione	Data di rilascio	Informazioni
1.0.1.1037	19 Ottobre 2016	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. L' CodeDeploy agente per le istanze di Amazon Linux, RHEL e Ubuntu Server è stato aggiornato con la seguente modifica. Per le istanze di Windows Server, la versione più recente rimane la 1.0.1.998. Miglioramento: l'agente è ora in grado di determinare quale versione di Ruby è installata su un'istanza, per cui può richiamare lo script `codedeploy-agent` che utilizza tale versione.
1.0.1.1011.1	17 agosto 2016	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Miglioramenti: rimosse le modifiche introdotte dalla versione 1.0.1.1011 a causa di problemi con il supporto di shell. Questa versione dell'agente è equivalente dal punto di vista funzional e alla versione 1.0.1.998 rilasciata l'11 luglio 2016.

Versione	Data di rilascio	Informazioni
1.0.1.1011	15 agosto 2016	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. L' CodeDeploy agente per le istanze di Amazon Linux, RHEL e Ubuntu Server è stato aggiornato con le seguenti modifiche . Per le istanze di Windows Server, la versione più recente rimane la 1.0.1.998. Funzionalità: è stato aggiunto il supporto per richiamare l' CodeDeploy agente utilizzando la shell bash sui sistemi operativi in cui è in uso il sistema init systemd. Miglioramento: è stato abilitato il supporto per tutte le versioni di Ruby 2.x nell'agente e nell'agente di aggiornamento. CodeDeploy CodeDeploy CodeDeploy Gli agenti aggiornat i non dipendono più solo da Ruby 2.0. (Ruby 2.0 è ancora necessario per le versioni deb e rpm del programma di installazione dell' CodeDeploy agente.)
1.0.1.998	11 luglio 2016	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Miglioramento: supporto fisso per l'esecuzione dell' CodeDeploy agente con profili utente diversi da root. La variabile `USER` è stata sostituita da `CODEDEPLOY_USER` per evitare conflitti con le variabili di ambiente.

Versione	Data di rilascio	Informazioni
1.0.1.966	16 giugno 2016	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Funzionalità: è stato introdotto il supporto per l'esecuzione dell' CodeDeploy agente con profili utente diversi da root. Miglioramento: risolto il supporto per specificare il numero di revisioni dell'applicazione che l'agente CodeDeploy può archiviare per un gruppo di distribuzione. Miglioramento: reso l' CodeDeploy agente compatibile con la versione 2.3 di AWS SDK per Ruby (aws-sdk-core 2.3). Miglioramento: sono stati risolti i problemi di codifica durante le distribuzioni. UTF-8 Miglioramento: maggiore precisione durante l'identificazione dei nomi dei processi.
1.0.1.950	24 marzo 2016	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Funzionalità: aggiunto il supporto proxy durante l'installazione. Miglioramento: è stato aggiornato lo script di installazione per non scaricare l' CodeDeploy agente se è già installata la versione più recente.
1.0.1.934	11 febbraio 2016	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Funzionalità: introdotto il supporto per specificare il numero di revisioni dell'applicazione che l'agente CodeDeploy può archiviare per un gruppo di distribuzione.

Versione	Data di rilascio	Informazioni
1.0.1.880	11 gennaio 2016	Nota: questa versione non è più supportata e le distribuzioni potrebbero avere esito negativo. Miglioramento: reso l' CodeDeploy agente compatibile con la versione 2.2 di AWS SDK per Ruby (aws-sdk-core 2.2). La versione 2.1.2 è ancora supportata.
1.0.1.854	17 Novembre 2015	Nota: questa versione non è più supportata. Se usi questa versione, le distribuzioni potrebbero avere esito negativo. Funzionalità: introdotto il supporto per l'algoritmo hash. SHA-256 Funzionalità: introdotto il supporto del monitoraggio delle versioni nei file `.version`. Funzionalità: l'ID del gruppo di distribuzione è ora disponibile mediante l'uso di una variabile di ambiente. Miglioramento: è stato aggiunto il supporto per il monitoraggio dei log degli CodeDeploy agenti tramite Amazon Logs. CloudWatch

Per informazioni correlate, consulta quanto segue:

- Determina la versione dell'agente CodeDeploy
- Installa l' CodeDeploy agente

Per una cronologia delle versioni degli CodeDeploy agenti, consulta l'archivio Release su. GitHub

Gestione del processo CodeDeploy

L' CodeDeploy agente utilizza systemd per gestire il processo dell'agente su tutte le distribuzioni Linux supportate.

Per la versione 2.0.x e successive, l'agente fornisce solo un'unità systemd nativa (). `codedeploy-agent.service` Non esiste uno script. `/etc/init.d/codedeploy-agent` Si usa `systemctl` per gestire l'agente o richiamare il file binario direttamente con`start`, `stop``restart`, o `status` sottocomandi.

Per la versione 1.8.x e precedenti, entrambe le distribuzioni rpm e deb vengono fornite con gli script di avvio all'indirizzo. `/etc/init.d/codedeploy-agent` Sui sistemi che supportano systemd, l'uso del `service` comando può eseguire inavvertitamente gli script init.d invece di consentire a systemd di gestire il processo. Utilizzatelo per le operazioni degli agenti `systemctl` per evitare questo problema.

Per gestire l' CodeDeploy agente, usa l'`systemctl`utilità:

```
sudo systemctl start codedeploy-agent
sudo systemctl stop codedeploy-agent
sudo systemctl restart codedeploy-agent
sudo systemctl status codedeploy-agent
```

Su macOS, l'agente viene eseguito come demone launchd`com.amazon.aws.codedeploy-agent`, definito in`/Library/LaunchDaemons/com.amazon.aws.codedeploy-agent.plist`. launchd avvia l'agente all'avvio e lo riavvia in caso di arresto anomalo. Per gestire l'agente su macOS, usa l'`launchctl`utilità:

```
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
```

Questi comandi avviano, interrompono, riavviano e mostrano rispettivamente lo stato dell'agente.

Revisione dell'applicazione e pulizia dei file di registro

L' CodeDeploy agente archivia le revisioni e i file di registro sulle istanze. L' CodeDeploy agente pulisce questi artefatti per risparmiare spazio su disco.

Registri di distribuzione delle revisioni delle applicazioni: è possibile utilizzare l'opzione:max_revisions: nel file di configurazione dell'agente per specificare il numero di revisioni

dell'applicazione da archiviare inserendo un numero intero positivo. CodeDeploy archivia anche i file di registro per tali revisioni. Tutti gli altri vengono eliminati, a eccezione del file di log dell'ultima distribuzione con esito positivo. Quel file di log viene sempre conservato, anche se il numero di distribuzioni con esito negativo supera il numero di revisioni conservate. Se non viene specificato alcun valore, CodeDeploy conserva le cinque revisioni più recenti oltre alla revisione attualmente distribuita.

CodeDeploy log: per le istanze Amazon Linux, Ubuntu Server, RHEL e macOS, l' CodeDeploy agente ruota i file di registro all'interno della cartella. `/var/log/aws/codedeploy-agent` Il file di registro viene ruotato ogni giorno alle 00:00:00 (ora dell'istanza) o quando il file raggiunge i 64 MB, a seconda dell'evento che si verifica per primo. Se si verifica una seconda rotazione lo stesso giorno, l'agente aggiunge un suffisso numerico (ad esempio,). `codedeploy-agent.20260715.1.log` L'agente elimina i file di registro dopo sette giorni. Il modello di denominazione per i file di log ruotati è `codedeploy-agent.`*`YYYYMMDD`*`.log`.

File installati dall'agente CodeDeploy

L' CodeDeploy agente archivia le revisioni, la cronologia di distribuzione e gli script di distribuzione nella directory principale di un'istanza. Il nome e la posizione predefiniti di questa directory sono:

`'/opt/codedeploy-agent/deployment-root'`per istanze Amazon Linux, Ubuntu Server, RHEL e macOS.

`'C:\ProgramData\Amazon\CodeDeploy'`per istanze di Windows Server.

È possibile utilizzare l'impostazione root_dir nel file di configurazione dell' CodeDeploy agente per configurare il nome e la posizione della directory. Per ulteriori informazioni, consulta CodeDeploy riferimento alla configurazione dell'agente.

Di seguito è riportato un esempio del file e della struttura della directory nella directory principale. La struttura presuppone che vi sia un numero N di gruppi di distribuzione e che ogni gruppo di distribuzione includa un numero N di distribuzioni.

```
|--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
```

- Le cartelle ID gruppo di distribuzione rappresentano ognuno dei gruppi di distribuzione. Il nome della directory del gruppo di distribuzione è il relativo ID (ad esempio, `acde1916-9099-7caf-fd21-012345abcdef`). Ciascuna directory del gruppo di distribuzione contiene una sottodirectory per ogni tentativo di distribuzione in quel gruppo di distribuzione.

  È possibile utilizzare il comando batch-get-deployments per trovare un ID del gruppo di distribuzione.

- Le cartelle ID distribuzione rappresentano ciascuna distribuzione in un gruppo di distribuzione. Il nome della directory di ciascuna distribuzione è il relativo ID. Ogni cartella contiene:
  - bundle.tar, un file compresso con i contenuti della revisione della distribuzione. Usa un'utilità di decompressione zip se desideri visualizzare la revisione.
  - deployment-archive, una directory che include i contenuti della revisione della distribuzione.
  - log, una directory che contiene un file `scripts.log`. Questo file elenca l'output di tutti gli script specificati nel file di distribuzione. AppSpec

  Se desideri trovare la cartella per una distribuzione ma non conosci l'ID di distribuzione o l'ID del gruppo di distribuzione, puoi utilizzare la AWS CodeDeploy console o AWS CLI trovarli. Per ulteriori informazioni, consulta Visualizza i dettagli CodeDeploy della distribuzione.

  Per impostazione predefinita, il numero massimo di distribuzioni che possono essere archiviate in un gruppo di distribuzione è cinque. Quando viene raggiunto il numero desiderato, le distribuzioni future vengono archiviate e l'archivio meno recente viene eliminato. È possibile utilizzare l'impostazione max_revisions nel file di configurazione dell' CodeDeploy agente per modificare l'impostazione predefinita. Per ulteriori informazioni, consulta CodeDeploy riferimento alla configurazione dell'agente.

  > **Note**
  >
  > Se desideri recuperare spazio nell'hard disk occupato dalle distribuzioni archiviate, imposta max_revisions su un valore basso, ad esempio 1 o 2. La distribuzione successiva elimina le distribuzioni archiviate in modo che il numero sia uguale al numero specificato.

- deployment-instructions contiene quattro file di testo per ogni gruppo di distribuzione:
  - [Deployment Group ID] -cleanup, un file di testo con una versione di annullamento di ogni comando eseguito durante una distribuzione. Un esempio di file è `acde1916-9099-7caf-fd21-012345abcdef-cleanup`.

- [Deployment Group ID]-install.json, un file JSON creato durante la distribuzione più recente. Contiene i comandi eseguiti durante la distribuzione. Un esempio di file è `acde1916-9099-7caf-fd21-012345abcdef-install.json`.
- [Deployment Group ID] _last_successful_install, un file di testo che elenca la directory di archivio dell'ultima distribuzione riuscita. Questo file viene creato quando l' CodeDeploy agente ha copiato nell'istanza tutti i file dell'applicazione di distribuzione. Viene utilizzato dall' CodeDeploy agente durante la distribuzione successiva per determinare quali `ApplicationStop` `BeforeInstall` script eseguire. Un esempio di file è `acde1916-9099-7caf-fd21-012345abcdef_last_successful_install`.
- [Deployment Group ID]_most_recent_install, un file di testo che scrive il nome della directory di archivio della distribuzione più recente. Questo file viene creato quando i file nella distribuzione vengono scaricati con successo. Il file [ID del gruppo di distribuzione] _last_successful_install viene creato dopo questo file, quando i file scaricati vengono copiati nella destinazione finale. Un esempio di file è `acde1916-9099-7caf-fd21-012345abcdef_most_recent_install`.

- deployment-logs contiene i seguenti file di log:
  - I file codedeploy-agent.yyyymmdd.log vengono creati per ogni giorno in cui avviene una distribuzione. Ogni file di log contiene informazioni sulle distribuzioni del giorno. Questi file di log possono essere utilizzati per problemi di debug come quelli relativi alle autorizzazioni. Il file di log viene inizialmente denominato `codedeploy-agent.log`. Il giorno successivo, la data delle distribuzioni viene inserita nel nome del file. Ad esempio, se oggi è il 3 gennaio 2018, è possibile vedere le informazioni su tutte le distribuzioni di oggi in `codedeploy-agent.log`. Domani, 4 gennaio 2018, il file di log verrà rinominato `codedeploy-agent.20180103.log`.
  - codedeploy-agent-deployments.log compila i contenuti dei file `scripts.log` per ogni distribuzione. I file `scripts.log` vengono posizionati nella sottocartella `logs` nella cartella `Deployment ID`. Le voci in questo file sono precedute da un ID di distribuzione. Ad esempio, "`[d-ABCDEF123]LifecycleEvent - BeforeInstall`" potrebbe essere scritto durante una distribuzione con un ID di `d-ABCDEF123`. Quando `codedeploy-agent-deployments.log` raggiunge la dimensione massima, l' CodeDeploy agente continua a scrivervi mentre elimina i vecchi contenuti.

Gestione delle operazioni degli CodeDeploy agenti

Le istruzioni in questa sezione mostrano come installare, disinstallare, reinstallare o aggiornare l' CodeDeploy agente e come verificare che l' CodeDeploy agente sia in esecuzione.

> **Tip**
>
> Alcune istruzioni in questa pagina variano a seconda della versione dell'agente. Se non sei sicuro della versione installata, esegui `/opt/codedeploy-agent/bin/codedeploy-agent --version` (Linux o macOS) `& 'C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe' --version` o (Windows). Se il comando non viene riconosciuto, stai utilizzando la versione 1.8.x o precedente. Per ulteriori informazioni, consulta Determina la versione dell'agente CodeDeploy.

Argomenti

- Verifica che l' CodeDeploy agente sia in esecuzione
- Determina la versione dell'agente CodeDeploy
- Installa l' CodeDeploy agente
- Aggiorna l' CodeDeploy agente
- Disinstalla l' CodeDeploy agente
- Invia i log degli CodeDeploy agenti a CloudWatch

Verifica che l' CodeDeploy agente sia in esecuzione

Questa sezione descrive i comandi da eseguire se sospetti che l' CodeDeploy agente abbia smesso di funzionare su un'istanza.

Argomenti

- Verifica che l' CodeDeploy agente per Amazon Linux o RHEL sia in esecuzione
- Verifica che l' CodeDeploy agente per Ubuntu Server sia in esecuzione
- Verifica che l' CodeDeploy agente per Windows Server sia in esecuzione
- Verifica che l' CodeDeploy agente per macOS sia in esecuzione

Verifica che l' CodeDeploy agente per Amazon Linux o RHEL sia in esecuzione

Per verificare se l' CodeDeploy agente è installato e in esecuzione, accedi all'istanza ed esegui il comando seguente:

```
systemctl status codedeploy-agent
```

Se il comando restituisce un errore, l' CodeDeploy agente non viene installato. Installarlo come descritto in Installa l' CodeDeploy agente per Amazon Linux o RHEL.

Se l' CodeDeploy agente è installato e in esecuzione, dovrebbe apparire un messaggio del tipo `The AWS CodeDeploy agent is running.` (codice di uscita 0).

Se vedi un messaggio del tipo `The AWS CodeDeploy agent is not running.` (codice di uscita 3), avvia il servizio ed esegui i due comandi seguenti, uno alla volta:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Per la versione 2.0.x e successive, il `codedeploy-agent status` comando restituisce i codici di LSB-compatible uscita: `0` (in esecuzione), `3` (non in esecuzione), `4` (sconosciuto). Durante l'uso`systemctl status`, viene visualizzato lo stato dell'unità (ad esempio,`Active: active (running)`).

Verifica che l' CodeDeploy agente per Ubuntu Server sia in esecuzione

Per verificare se l' CodeDeploy agente è installato e in esecuzione, accedi all'istanza ed esegui il comando seguente:

```
systemctl status codedeploy-agent
```

Se il comando restituisce un errore, l' CodeDeploy agente non viene installato. Installarlo come descritto in Installa l' CodeDeploy agente per Ubuntu Server.

Se l' CodeDeploy agente è installato e in esecuzione, dovrebbe apparire un messaggio del tipo `The AWS CodeDeploy agent is running.` (codice di uscita 0).

Se vedi un messaggio del tipo `The AWS CodeDeploy agent is not running.` (codice di uscita 3), avvia il servizio ed esegui i due comandi seguenti, uno alla volta:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Verifica che l' CodeDeploy agente per Windows Server sia in esecuzione

Per verificare se l' CodeDeploy agente è installato e in esecuzione, accedi all'istanza ed esegui il comando seguente:

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Verrà visualizzato un output simile al seguente:

```
Status   Name               DisplayName
------   ----               -----------
Running codedeployagent    CodeDeploy Host Agent Service
```

Se il comando restituisce un errore, l' CodeDeploy agente non viene installato. Installarlo come descritto in Installa l' CodeDeploy agente per Windows Server.

Se `Status` mostra un valore diverso da `Running`, avviare il servizio con il comando seguente:

```
powershell.exe -Command Start-Service -Name codedeployagent
```

È possibile riavviare il servizio con il comando seguente:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

È possibile arrestare il servizio con il comando seguente:

```
powershell.exe -Command Stop-Service -Name codedeployagent
```

Verifica che l' CodeDeploy agente per macOS sia in esecuzione

Per verificare se l' CodeDeploy agente è installato e in esecuzione, accedi all'istanza ed esegui il comando seguente:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Se il comando restituisce un errore perché il file non viene trovato, l' CodeDeploy agente non viene installato. Installarlo come descritto in Installa l' CodeDeploy agente per macOS.

Se l' CodeDeploy agente è installato e in esecuzione, dovrebbe apparire un messaggio del tipo The `AWS CodeDeploy agent is running.` (codice di uscita 0).

Se vedi un messaggio del tipo `The AWS CodeDeploy agent is not running.` (codice di uscita 3), avvia l'agente con il seguente comando:

```
sudo launchctl kickstart system/com.amazon.aws.codedeploy-agent
```

Se il comando segnala che il servizio non è stato trovato, il demone launchd non viene caricato. Caricatelo e avviatelo con il seguente comando:

```
sudo launchctl bootstrap system /Library/LaunchDaemons/com.amazon.aws.codedeploy-
agent.plist
```

Per vedere lo stato del demone launchd, esegui:

```
sudo launchctl print system/com.amazon.aws.codedeploy-agent
```

Determina la versione dell'agente CodeDeploy

Puoi determinare la versione dell' CodeDeploy agente in esecuzione sull'istanza in due modi.

Innanzitutto, per la versione 2.0.x e successive, utilizza il `codedeploy-agent --version` comando su tutte le piattaforme.

In secondo luogo, è possibile visualizzare la versione in un `.version` file sull'istanza. Su Amazon Linux, RHEL, Ubuntu Server e macOS il file si trova `/opt/codedeploy-agent/.version` in, e su Windows Server in. `C:\ProgramData\Amazon\CodeDeploy\.version` Il contenuto del file varia in base alla versione:

- Per la versione 2.0.x e successive, il file contiene solo il numero di versione, nel formato. `agent_version: 2.0.0` Questo valore corrisponde alla versione che l'agente segnala al servizio. CodeDeploy
- Per la versione 1.8.x e precedenti, la stringa della versione utilizza il formato `OFFICIAL_1.0.1.854_rpm` (o_deb,_msi).

Argomenti

- Determina la versione su Amazon Linux o RHEL

- Determina la versione su Ubuntu Server
- Determina la versione su Windows Server
- Determina la versione su macOS
- Determina l'ultima versione dell'agente disponibile in un Regione AWS

Determina la versione su Amazon Linux o RHEL

Accedere all'istanza ed eseguire il comando seguente:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Per le versioni precedenti alla 2.0.0, puoi anche utilizzare:

```
sudo yum info codedeploy-agent
```

Determina la versione su Ubuntu Server

Accedere all'istanza ed eseguire il comando seguente:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Per le versioni precedenti alla 2.0.0, puoi anche usare:

```
sudo dpkg -s codedeploy-agent
```

Determina la versione su Windows Server

Accedere all'istanza ed eseguire il comando seguente:

```
& 'C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe' --version
```

Determina la versione su macOS

Accedere all'istanza ed eseguire il comando seguente:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Puoi anche vedere la versione del pacchetto installato:

```
pkgutil --pkg-info com.amazon.aws.codedeploy-agent
```

Determina l'ultima versione dell'agente disponibile in un Regione AWS

Per scoprire qual è la versione più recente disponibile nella regione in Regione AWS cui esegui la distribuzione, consulta il bucket del CodeDeploy Resource Kit per quella regione. Ogni bucket regionale contiene un `latestv2/LATEST_VERSION` file che nomina i pacchetti di agenti attualmente rilasciati in quella regione. Il file è leggibile pubblicamente, quindi non sono necessarie credenziali per recuperarlo.

```
curl https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/LATEST_VERSION
```

Sostituisci *`bucket-name`* e *`region-identifier`* con il bucket del kit di risorse e l'identificatore di regione per la regione in cui esegui la distribuzione. Per l'elenco, consulta Nomi dei bucket del Resource Kit per regione.

Il comando restituisce un output simile al seguente:

```
{
    "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"
}
```

Ogni chiave indica un tipo di pacchetto e, per RPM e DEB, anche l'architettura dell'istanza. Ogni valore è la chiave oggetto Amazon S3 di quel pacchetto e la versione dell'agente e il numero di build fanno parte del nome del file, nel formato. *`agent-version`*-*`build-number`* Nell'output precedente, l'ultima versione disponibile nella regione è la 2.1.0.

Per la versione 2.1.2 e successive, il file ha anche una `pkg_aarch64` chiave, che nomina il pacchetto di installazione di macOS.

Per installare o aggiornare alla versione denominata questo file, consulta e. Installa l' CodeDeploy agente Aggiorna l' CodeDeploy agente Per le funzionalità incluse in ogni versione, vedereCronologia delle versioni dell'agente CodeDeploy.

Installa l' CodeDeploy agente

Per utilizzarlo CodeDeploy su istanze EC2 o server locali, l' CodeDeploy agente deve essere prima installato. È possibile installare l' CodeDeploy agente con AWS Systems Manager o con la riga di comando. Per ulteriori informazioni su Systems Manager, vedere What is AWS Systems Manager.

Per la versione consigliata da installare, vedereCronologia delle versioni dell'agente CodeDeploy.

Argomenti

- Installa l' CodeDeploy agente utilizzando AWS Systems Manager
- Installa l' CodeDeploy agente utilizzando la riga di comando

Installa l' CodeDeploy agente utilizzando AWS Systems Manager

Puoi usare Console di gestione AWS o il AWS CLI per installare l' CodeDeploy agente sulle tue istanze Amazon EC2 o locali utilizzando. AWS Systems ManagerÈ possibile scegliere di installare una versione specifica o di installare sempre la versione più recente dell'agente. Per ulteriori informazioni su AWS Systems Manager, consulta What is. AWS Systems Manager

Puoi anche installare l' CodeDeploy agente da un bucket Amazon S3. Per informazioni sull'uso di un collegamento per il download di Amazon S3, consulta Installa l' CodeDeploy agente utilizzando la riga di comando.

Argomenti

- Prerequisiti
- Installa l' CodeDeploy agente

Prerequisiti

Segui i passaggi indicati Guida introduttiva con CodeDeploy per configurare le autorizzazioni IAM e il. AWS CLI

Se si installa l' CodeDeploy agente su un server locale con Systems Manager, è necessario registrare il server locale con Amazon EC2 Systems Manager. Per ulteriori informazioni, vedere Configurazione di Systems Manager in ambienti ibridi nella Guida AWS Systems Manager per l'utente.

Installa l' CodeDeploy agente

Prima di poter utilizzare Systems Manager per installare l' CodeDeploy agente, è necessario assicurarsi che l'istanza sia configurata correttamente per Systems Manager.

Installazione o aggiornamento dell'agente SSM

Su un'istanza Amazon EC2, l' CodeDeploy agente richiede che l'istanza esegua la versione 2.3.274.0 o successiva. Prima di installare l' CodeDeploy agente, aggiorna o installa l'agente SSM sull'istanza, se non l'hai già fatto.

L'agente SSM è preinstallato su alcune AMI Amazon EC2 fornite da. AWS Per ulteriori informazioni, consulta Amazon Machine Images (AMI) con agente SSM preinstallato.

> **Note**
>
> Assicurati che anche il sistema operativo dell'istanza sia supportato dall'agente. CodeDeploy Per ulteriori informazioni, consulta Sistemi operativi supportati dall'agente CodeDeploy.

Per informazioni sull'installazione o l'aggiornamento dell'agente SSM su un'istanza che esegue Linux, consulta Installazione e configurazione dell'agente SSM sulle istanze Linux nella Guida per l'utente.AWS Systems Manager

Per informazioni sull'installazione o l'aggiornamento dell'agente SSM su un'istanza che esegue Windows Server, consulta Installazione e configurazione dell'agente SSM sulle istanze Windows nella Guida per l'utente.AWS Systems Manager

Per informazioni sull'installazione o l'aggiornamento dell'agente SSM su un'istanza EC2 Mac, consulta Installazione e disinstallazione manuale dell'agente SSM sulle istanze EC2 per macOS nella Guida per l'utente.AWS Systems Manager

(Facoltativo) Verifica dei prerequisiti di Systems Manager

Prima di utilizzare Systems Manager Run Command per installare l' CodeDeploy agente, verificate che le istanze soddisfino i requisiti minimi di Systems Manager. Per ulteriori informazioni, consulta Setting up AWS Systems Manager in AWS Systems Manager User Guide.

Installa l'agente CodeDeploy

Con SSM, puoi installarlo CodeDeploy una sola volta o impostare una pianificazione per installare nuove versioni.

Per installare l' CodeDeploy agente, scegli il pacchetto Systems Manager Distributor per la versione dell'agente desiderata, quindi segui i passaggi in Installare o aggiornare i pacchetti con il AWS Systems Manager distributore. Il pacchetto si differenzia in base alla versione dell'agente:

- Per la versione 2.0.x e successive, scegli il pacchetto. `AWSCodeDeployAgentV2` Questo è l'unico pacchetto che supporta macOS, a partire dalla versione 2.1.2.
- Per la versione 1.8.x e precedenti, scegli il pacchetto. `AWSCodeDeployAgent`

Installa l' CodeDeploy agente utilizzando la riga di comando

Utilizzare i seguenti argomenti per installare ed eseguire l' CodeDeploy agente tramite la riga di comando.

Argomenti

- Installa l' CodeDeploy agente per Amazon Linux o RHEL
- Installa l' CodeDeploy agente per Ubuntu Server
- Installa l' CodeDeploy agente per Windows Server
- Installa l' CodeDeploy agente per macOS

Installa l' CodeDeploy agente per Amazon Linux o RHEL

Accedere all'istanza ed eseguire i comandi seguenti, uno alla volta. L'esecuzione `sudo yum update` iniziale del comando è considerata una procedura consigliata `yum` per l'installazione dei pacchetti, ma è possibile ignorarla se non si desidera aggiornare tutti i pacchetti.

```
sudo yum update
```

Per la versione 2.0.x e successive:

```
sudo yum install wget
```

Per la versione 1.8.x e precedenti:

```
sudo yum install ruby wget
```

(Facoltativo) Per pulire l'AMI da tutte le informazioni di memorizzazione nella cache degli agenti precedenti, esegui lo script seguente:

```
#!/bin/bash
CODEDEPLOY_BIN="/opt/codedeploy-agent/bin/codedeploy-agent"
$CODEDEPLOY_BIN stop
yum erase codedeploy-agent -y
```

Passa alla tua home directory:

```
cd /home/ec2-user
```

> **Note**
>
> Nel comando precedente, `/home/ec2-user` rappresenta il nome utente predefinito per un'istanza Amazon Linux o RHEL Amazon EC2. Se l'istanza è stata creata utilizzando un'AMI personalizzata, il proprietario dell'AMI potrebbe avere specificato un nome utente predefinito diverso.

Scarica il programma di installazione dell' CodeDeploy agente. Il `latestv2/` prefisso serve il programma di installazione della versione 2.0.x e il `latest/` prefisso serve il programma di installazione della versione 1.8.x.

Per la versione 2.0.x e successive:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/install
```

Per la versione 1.8.x e precedenti:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```

*`bucket-name`*è il nome del bucket Amazon S3 che contiene i file CodeDeploy Resource Kit per la tua regione ed *`region-identifier`* è l'identificatore della tua regione.

Ad esempio, per la versione 2.0.x e successive:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/install`

Per la versione 1.8.x e precedenti:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install`

Per un elenco dei nomi dei bucket e degli identificatori di regione, consulta. Nomi dei bucket del Resource Kit per regione

Imposta i permessi di esecuzione sul file: `install`

```
chmod +x ./install
```

Per installare la versione più recente dell' CodeDeploy agente:

-
```
sudo ./install auto
```

Per installare una versione specifica dell' CodeDeploy agente:

- Elenca le versioni disponibili nella tua regione:

  ```
  aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier
   | grep '\.rpm$'
  ```
- Installa una delle versioni:

  Per la versione 2.0.x e successive, i pacchetti sono specifici dell'architettura:

  ```
  sudo ./install auto -v releases/codedeploy-agent-version.arch.rpm
  ```

  Dov'è `x86_64` o `aarch64` dipende dall'*arch*architettura dell'istanza.

  Per la versione 1.8.x e precedenti, i pacchetti utilizzano: `noarch`

  ```
  sudo ./install auto -v releases/codedeploy-agent-version.noarch.rpm
  ```

  Per l'ultima versione rilasciata, vedere. Cronologia delle versioni dell'agente CodeDeploy

Per verificare che il servizio sia in esecuzione, eseguire il comando seguente:

```
systemctl status codedeploy-agent
```

Per la versione 2.0.x e successive, se l'agente è in esecuzione, dovresti visualizzare un messaggio simile `The AWS CodeDeploy agent is running.`

Per la versione 1.8.x e precedenti, se l'agente è in esecuzione, dovresti vedere un messaggio simile `The AWS CodeDeploy agent is running.`

Se l'agente non è in esecuzione, avvia il servizio e controlla lo stato:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Installa l' CodeDeploy agente per Ubuntu Server

Per installare l' CodeDeploy agente su Ubuntu Server

1. Accedi all'istanza.
2. Inserisci i seguenti comandi, uno dopo l'altro:

   ```
   sudo apt update
   ```

   Per la versione 2.0.x e successive:

   ```
   sudo apt install wget
   ```

   Per la versione 1.8.x e precedenti:

   ```
   sudo apt install ruby-full wget
   ```
3. Immetti il comando seguente:

   ```
   cd /home/ubuntu
   ```

   */home/ubuntu*rappresenta il nome utente predefinito per un'istanza di Ubuntu Server. Se l'istanza è stata creata utilizzando un'AMI personalizzata, il proprietario dell'AMI potrebbe avere specificato un nome utente predefinito diverso.
4. Inserire il seguente comando. Il `latestv2/` prefisso serve il programma di installazione della versione 2.0.x e il `latest/` prefisso serve il programma di installazione della versione 1.8.x.

   Per la versione 2.0.x e successive:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/install
```

Per la versione 1.8.x e precedenti:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```

*bucket-name*è il nome del bucket Amazon S3 che contiene i file CodeDeploy Resource Kit per la tua regione ed *region-identifier* è l'identificatore della tua regione.

Ad esempio, per la versione 2.0.x e successive:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/install`

Per la versione 1.8.x e precedenti:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install`

Per un elenco dei nomi dei bucket e degli identificatori di regione, consulta. Nomi dei bucket del Resource Kit per regione

5. Immetti il comando seguente:

```
chmod +x ./install
```

6. Esegui una delle seguenti operazioni:

   - Per installare la versione più recente dell' CodeDeploy agente:

     ```
     sudo ./install auto
     ```

   - Per installare una versione specifica dell' CodeDeploy agente:

     - Elenca le versioni disponibili nella tua regione:

       ```
       aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier | grep '\.deb$'
       ```

     - Installa una delle versioni:

Per la versione 2.0.x e successive, i pacchetti sono specifici dell'architettura:

```
sudo ./install auto -v releases/codedeploy-agent_version_arch.deb
```

Dov'è amd64 o arm64 dipende dall'*arch*architettura dell'istanza.

Per la versione 1.8.x e precedenti, i pacchetti utilizzano una convenzione di denominazione diversa:

```
sudo ./install auto -v releases/codedeploy-agent_version_all.deb
```

Per l'ultima versione rilasciata, vedere. Cronologia delle versioni dell'agente CodeDeploy

Per verificare che il servizio sia in esecuzione

1. Immetti il comando seguente:

   ```
   systemctl status codedeploy-agent
   ```

   Se l' CodeDeploy agente è installato e in esecuzione:

   Per la versione 2.0.x e successive, dovresti vedere: `The AWS CodeDeploy agent is running.`

   Per la versione 1.8.x e precedenti, dovresti vedere: `The AWS CodeDeploy agent is running.`
2. Se l'agente non è in esecuzione, avvia il servizio e verifica lo stato:

   ```
   systemctl start codedeploy-agent
   ```

   ```
   systemctl status codedeploy-agent
   ```

Installa l' CodeDeploy agente per Windows Server

Nelle istanze di Windows Server, puoi utilizzare uno di questi metodi per scaricare e installare l' CodeDeploy agente:

- Utilizzo AWS Systems Manager (consigliato)
- Esegui una serie di PowerShell comandi di Windows.
- Selezionare un collegamento diretto per il download.
- Esegui un comando di copia Amazon S3.

> **Note**
>
> La cartella in cui è installato l' CodeDeploy agente è`C:\ProgramData\Amazon\CodeDeploy`. Non collocate giunzioni di directory o collegamenti simbolici su questo percorso. Per la versione 2.0.x e successive, l'agente impone questo requisito e si rifiuta di avviarsi se il file binario del servizio non è inattivo. `C:\ProgramData\Amazon\CodeDeploy\`

Argomenti

- Utilizzo di Systems Manager
- Usa Windows PowerShell
- Uso di un collegamento diretto
- Usa un comando di copia Amazon S3

Utilizzo di Systems Manager

Segui le istruzioni riportate per installare l'agenteInstalla l' CodeDeploy agente utilizzando AWS Systems Manager. CodeDeploy

Usa Windows PowerShell

Accedi all'istanza ed esegui i seguenti comandi in Windows PowerShell:

1. Richiedere che tutti gli script e i file di configurazione scaricati da Internet siano firmati da un autore attendibile. Se viene richiesto di modificare la policy di esecuzione, digitare "**Y**".

   ```
   Set-ExecutionPolicy RemoteSigned
   ```

2. Carica il AWS Tools for Windows PowerShell.

```
Import-Module AWSPowerShell
```

3. Crea una directory in cui scaricare il file di installazione dell' CodeDeploy agente.

```
New-Item -Path "c:\temp" -ItemType "directory" -Force
```

4. Configura AWS le credenziali utilizzando i `Initialize-AWSDefaultConfiguration` comandi `Set-AWSCredential` and. Per ulteriori informazioni, vedere Utilizzo AWS delle credenziali negli AWS strumenti per la guida per PowerShell l'utente.
5. Scaricate il file di installazione dell' CodeDeploy agente.

> **Note**
>
> Per l'ultima versione rilasciata, vedereCronologia delle versioni dell'agente CodeDeploy.

Per installare la versione più recente dell' CodeDeploy agente. Il `latestv2/` prefisso serve il programma di installazione della versione 2.x (2.0.x e successive) e il `latest/` prefisso serve il programma di installazione della versione 1.8.x.

- Per la versione 2.0.x e successive:

```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latestv2/
codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi
```

Per la versione 1.8.x e precedenti:

```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latest/
codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi
```

Per installare una versione specifica dell' CodeDeployagente:

-
```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key releases/
codedeploy-agent-###.msi -File c:\temp\codedeploy-agent.msi
```

*bucket-name*è il nome del bucket Amazon S3 che contiene i file CodeDeploy Resource Kit per la tua regione. Ad esempio, per la regione Stati Uniti orientali (Ohio), sostituisci con. *bucket-name* `aws-codedeploy-us-east-2` Per un elenco dei nomi dei bucket, vedi. Nomi dei bucket del Resource Kit per regione

6. Esegui il file di installazione dell' CodeDeploy agente.

```
c:\temp\codedeploy-agent.msi /quiet /l c:\temp\host-agent-install-log.txt
```

Per verificare che il servizio sia in esecuzione, eseguire il comando seguente:

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Se l' CodeDeploy agente è stato appena installato e non è stato avviato, dopo aver eseguito il Get-Service comando, in Stato, dovresti vedere**`Start...`**:

```

Status     Name               DisplayName
------     ----               -----------
Start...   codedeployagent    CodeDeploy Host Agent Service
```

Se l' CodeDeploy agente è già in esecuzione, dopo aver eseguito il Get-Service comando, in Status, dovresti vedere**`Running`**:

```

Status     Name               DisplayName
------     ----               -----------
Running    codedeployagent    CodeDeploy Host Agent Service
```

Uso di un collegamento diretto

Se le impostazioni di sicurezza del browser sull'istanza di Windows Server forniscono le autorizzazioni (ad esempio, per`https://s3.*.amazonaws.com`), puoi utilizzare un collegamento diretto alla tua regione per scaricare l' CodeDeploy agente ed eseguire il programma di installazione manualmente.

Il `latestv2/` prefisso serve il programma di installazione della versione 2.x (2.0.x e successive) e il prefisso serve il programma di installazione della `latest/` versione 1.8.x.

Per la versione 2.0.x e successive, il collegamento è:

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latestv2/codedeploy-agent.msi
```

Per la versione 1.8.x e precedenti:

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latest/codedeploy-agent.msi
```

... *region* dov'è la AWS regione in cui stai distribuendo la tua applicazione.

Ad esempio, per la versione 2.0.x e successive:

```
https://s3.af-south-1.amazonaws.com/aws-codedeploy-af-south-1/latestv2/codedeploy-
agent.msi
```

> **Important**
>
> Ottieni il `.msi` file dalla stessa regione dell'applicazione. CodeDeploy La scelta di una regione diversa può causare `inconsistent region` errori nel `codedeploy-agent-log` file durante l'esecuzione del `.msi` file.

Usa un comando di copia Amazon S3

Se AWS CLI è installato sull'istanza, puoi utilizzare il comando Amazon S3 cp per scaricare l' CodeDeploy agente ed eseguire il programma di installazione manualmente. Per informazioni, consulta Installare il AWS Command Line Interface su Microsoft Windows.

Il `latestv2/` prefisso serve il programma di installazione della versione 2.x (2.0.x e successive) e il `latest/` prefisso serve il programma di installazione della versione 1.8.x.

Per la versione 2.0.x e successive, il comando Amazon S3 è:

```
aws s3 cp s3://aws-codedeploy-region/latestv2/codedeploy-agent.msi codedeploy-agent.msi
 --region region
```

Per la versione 1.8.x e precedenti:

```
aws s3 cp s3://aws-codedeploy-region/latest/codedeploy-agent.msi codedeploy-agent.msi
 --region region
```

... *region* dov'è la AWS regione in cui stai distribuendo la tua applicazione.

Ad esempio, per la versione 2.0.x e successive:

```
aws s3 cp s3://aws-codedeploy-af-south-1/latestv2/codedeploy-agent.msi codedeploy-agent.msi --region af-south-1
```

Installa l' CodeDeploy agente per macOS

Accedere all'istanza ed eseguire i comandi seguenti, uno alla volta.

Passa alla tua home directory:

```
cd ~
```

Scarica il pacchetto di installazione CodeDeploy dell'agente:

```
curl -O https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/codedeploy-agent.aarch64.pkg
```

*bucket-name*è il nome del bucket Amazon S3 che contiene i file CodeDeploy Resource Kit per la tua regione ed *region-identifier* è l'identificatore della tua regione.

Ad esempio:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/codedeploy-agent.aarch64.pkg`

Per un elenco dei nomi dei bucket e degli identificatori di regione, consulta. Nomi dei bucket del Resource Kit per regione

Per installare la versione più recente dell' CodeDeploy agente:

-
```
sudo installer -pkg codedeploy-agent.aarch64.pkg -target /
```

Per installare una versione specifica dell' CodeDeploy agente:

- Elenca le versioni disponibili nella tua regione:

```
aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier
 | grep '\.pkg$'
```

- Scarica e installa una delle versioni:

```
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 /
```

Per l'ultima versione rilasciata, vediCronologia delle versioni dell'agente CodeDeploy.

Per verificare che il servizio sia in esecuzione, eseguire il comando seguente:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Se l'agente è in esecuzione, dovresti vedere un messaggio simile `The AWS CodeDeploy agent is running.`

Se l'agente non è in esecuzione, avvia il servizio e controlla lo stato:

```
sudo launchctl kickstart system/com.amazon.aws.codedeploy-agent
```

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Aggiorna l' CodeDeploy agente

È possibile configurare gli aggiornamenti automatici e pianificati dell' CodeDeploy agente su tutti i sistemi operativi supportati utilizzando AWS Systems Manager. È inoltre possibile forzare gli aggiornamenti su tutti i sistemi operativi supportati mediante l'esecuzione di un comando su un'istanza.

Argomenti

- Aggiorna l' CodeDeploy agente su Amazon Linux o RHEL
- Aggiorna l' CodeDeploy agente su Ubuntu Server
- Aggiorna l' CodeDeploy agente su Windows Server

- Aggiorna l' CodeDeploy agente su macOS

Aggiorna l' CodeDeploy agente su Amazon Linux o RHEL

Per configurare gli aggiornamenti automatici e pianificati dell' CodeDeploy agente utilizzando AWS Systems Manager, segui la procedura descritta in Installare l' CodeDeploy agente con AWS Systems Manager.

Per la versione 2.0.x e successive, per forzare un aggiornamento, esegui:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

In alternativa, scarica nuovamente ed esegui lo script regionale. `latestv2/install` Per le istruzioni di installazione, consulta Installa l' CodeDeploy agente per Amazon Linux o RHEL.

Per la versione 1.8.x e precedenti, per forzare un aggiornamento, esegui:

```
sudo /opt/codedeploy-agent/bin/install auto
```

> **Important**
>
> Attualmente non esiste un percorso di aggiornamento automatico dalla versione 1.8.x alla 2.0.0. Per eseguire l'aggiornamento alla versione 2.0.0, installalo con il pacchetto `AWSCodeDeployAgentV2` Systems Manager Distributor o esegui lo script regionale `latestv2/install` manualmente. Per istruzioni, consulta l'argomento Installa l' CodeDeploy agente.

Aggiorna l' CodeDeploy agente su Ubuntu Server

Per configurare gli aggiornamenti automatici e pianificati dell' CodeDeploy agente utilizzando AWS Systems Manager, segui la procedura descritta in Installare l' CodeDeploy agente con AWS Systems Manager.

Per la versione 2.0.x e successive, per forzare un aggiornamento, esegui:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

In alternativa, scarica nuovamente ed esegui lo script regionale. `latestv2/install` Per le istruzioni di installazione, consulta Installa l' CodeDeploy agente per Ubuntu Server.

Per la versione 1.8.x e precedenti, per forzare un aggiornamento, esegui:

```
sudo /opt/codedeploy-agent/bin/install auto
```

> **Important**
>
> Attualmente non esiste un percorso di aggiornamento automatico dalla versione 1.8.x alla 2.0.0. Per eseguire l'aggiornamento alla versione 2.0.0, installalo con il pacchetto `AWSCodeDeployAgentV2` Systems Manager Distributor o esegui lo script regionale `latestv2/install` manualmente. Per istruzioni, consulta l'argomento Installa l' CodeDeploy agente.

Aggiorna l' CodeDeploy agente su Windows Server

È possibile abilitare gli aggiornamenti automatici dell' CodeDeploy agente con AWS Systems Manager. Con Systems Manager, puoi configurare una pianificazione di aggiornamento per le tue istanze Amazon EC2 o locali creando un'associazione con Systems Manager State Manager. Puoi anche aggiornare manualmente l' CodeDeploy agente disinstallando la versione corrente e installandone una più recente.

Argomenti

- Imposta l'aggiornamento automatico CodeDeploy dell'agente con AWS Systems Manager
- Aggiorna l' CodeDeploy agente manualmente

Imposta l'aggiornamento automatico CodeDeploy dell'agente con AWS Systems Manager

Per configurare Systems Manager e abilitare gli aggiornamenti automatici dell' CodeDeploy agente, seguire le istruzioni riportate in Installare l' CodeDeploy agente utilizzando AWS Systems Manager.

Aggiorna l' CodeDeploy agente manualmente

Per aggiornare l' CodeDeploy agente manualmente, è possibile installare la versione più recente dalla CLI o utilizzando Systems Manager. Segui le istruzioni in Installare l'agente CodeDeploy. Si consiglia di disinstallare le versioni precedenti dell' CodeDeploy agente seguendo le istruzioni riportate in Disinstallare l' CodeDeployagente.

Aggiorna l' CodeDeploy agente su macOS

Per configurare gli aggiornamenti automatici e pianificati dell' CodeDeploy agente utilizzando AWS Systems Manager, segui i passaggi indicati Installa l' CodeDeploy agente utilizzando AWS Systems Manager e scegli il `AWSCodeDeployAgentV2` pacchetto.

Per forzare un aggiornamento, esegui:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

In alternativa, scarica il `.pkg` file più recente e installalo sull'agente esistente. Per le istruzioni di installazione, consulta Installa l' CodeDeploy agente per macOS.

Disinstalla l' CodeDeploy agente

Puoi rimuovere l' CodeDeploy agente dalle istanze quando non è più necessario o quando desideri eseguire una nuova installazione.

Disinstalla l' CodeDeploy agente da Amazon Linux o RHEL

Per disinstallare l' CodeDeploy agente, accedi all'istanza ed esegui il seguente comando:

```
sudo yum erase codedeploy-agent
```

Disinstalla CodeDeploy l'agente da Ubuntu Server

Per disinstallare l' CodeDeploy agente, accedi all'istanza ed esegui il seguente comando:

```
sudo dpkg --purge codedeploy-agent
```

Disinstalla CodeDeploy l'agente da Windows Server

Per disinstallare l' CodeDeploy agente, accedi all'istanza ed esegui i seguenti tre comandi, uno alla volta:

```
wmic
product where name="CodeDeploy Host Agent" call uninstall /nointeractive
exit
```

> **Note**
>
> In Windows Server 2025 e versioni successive, `wmic` è obsoleto. Utilizzate invece il seguente comando: PowerShell
>
> ```
> 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 }
> ```

Puoi anche accedere all'istanza e, nel Pannello di controllo, aprire Programmi e funzionalità, scegliere CodeDeploy Host Agent e quindi scegliere Disinstalla.

Disinstalla l' CodeDeploy agente da macOS

Per disinstallare l' CodeDeploy agente, accedi all'istanza ed esegui il seguente comando:

```
sudo /opt/codedeploy-agent/bin/uninstall.sh
```

Invia i log degli CodeDeploy agenti a CloudWatch

Puoi inviare le metriche e i dati di registro dell' CodeDeploy agente CloudWatch utilizzando l' CloudWatch agente unificato o, più semplicemente, l'agente. CloudWatch

Utilizza le seguenti istruzioni per installare l' CloudWatch agente e configurarlo per l'uso con CodeDeploy gli agenti.

Prerequisiti

Prima di iniziare, completa le seguenti attività:

- Installa l' CodeDeploy agente e assicurati che sia in esecuzione. Per ulteriori informazioni, consultare Installa l' CodeDeploy agente e Verifica che l' CodeDeploy agente sia in esecuzione.
- Installa l' CloudWatch agente. Per ulteriori informazioni, vedere Installazione dell' CloudWatch agente.
- Aggiungi le seguenti autorizzazioni al profilo dell'istanza CodeDeploy IAM:
  - CloudWatchLogsFullAccess

- CloudWatchAgentServerPolicy

Per ulteriori informazioni sul profilo dell' CodeDeploy istanza, consulta Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2 diGuida introduttiva con CodeDeploy.

Configurare l' CloudWatch agente per raccogliere i CodeDeploy log

È possibile configurare l' CloudWatch agente eseguendo una procedura guidata o creando o modificando manualmente un file di configurazione.

Per configurare l' CloudWatch agente utilizzando la procedura guidata (Linux)

1. Esegui la procedura guidata, come descritto in Esecuzione della procedura guidata di configurazione dell' CloudWatch agente.
2. Nella procedura guidata, quando richiesto, inserisci. `Do you want to monitor any log files?` **`1`**
3. Specificate il file di registro CodeDeploy dell'agente, come segue:
   1. Per `Log file path` inserire il percorso del file di CodeDeploy registro, ad esempio:**`/var/log/aws/codedeploy-agent/codedeploy-agent.log`**.
   2. Per `Log group name` inserire il nome di un gruppo di log, ad esempio:**`codedeploy-agent-log`**.
   3. Per `Log stream name` inserire il nome di un flusso di log, per esempio:**`{instance_id}-codedeploy-agent-log`**.
4. Quando richiestoDo `you want to specify any additional log files?`, inserisci**`1`**.
5. Specificate i registri di distribuzione degli CodeDeploy agenti, come segue:
   1. Per `Log file path` inserire il percorso del file di registro di CodeDeploy distribuzione, ad esempio:**`/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log`**.
   2. Per `Log group name` inserire il nome di un gruppo di log, ad esempio:**`codedeploy-agent-deployment-log`**.
   3. Per `Log stream name` inserire il nome di un flusso di log, per esempio:**`{instance_id}-codedeploy-agent-deployment-log`**.
6. Quando richiestoDo `you want to specify any additional log files?`, inserisci**`1`**.
7. Specificate i registri CodeDeploy dell'agente di aggiornamento, come segue:

1. Per`Log file path`, immettete il percorso del file di registro dell' CodeDeploy aggiornamento. Il percorso dipende dalla versione dell'agente in uso:
   - Per la versione 2.0.x e successive, usa. **`/var/log/aws/codedeploy-agent/codedeploy-agent-updater.log`**
   - Per la versione 1.8.x e precedenti, usa. **`/tmp/codedeploy-agent.update.log`**
2. Per`Log group name` inserire il nome di un gruppo di log, ad esempio:. **`codedeploy-agent-updater-log`**
3. Per`Log stream name` inserire il nome di un flusso di log, per esempio:**`{instance_id}-codedeploy-agent-updater-log`**.

Per configurare l' CloudWatch agente utilizzando la procedura guidata (Windows)

1. Esegui la procedura guidata, come descritto in Esecuzione della procedura guidata di configurazione dell' CloudWatch agente.
2. Nella procedura guidata, quando richiesto, inserisci.`Do you want to monitor any customized log files?` **`1`**
3. Specificate il file di CodeDeploy registro nel modo seguente:
   1. Per`Log file path`, immettere il percorso del file di registro CodeDeploy dell'agente. Il percorso dipende dalla versione dell'agente:
      - Per la versione 2.0.x e successive, usa. **`C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log`**
      - Per la versione 1.8.x e precedenti, usa. **`C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt`**
   2. Per`Log group name` inserire il nome di un gruppo di log, ad esempio:. **`codedeploy-agent-log`**
   3. Per`Log stream name` inserire il nome di un flusso di log, per esempio:**`{instance_id}-codedeploy-agent-log`**.
4. Quando richiestoDo `you want to specify any additional log files?`, inserisci**`1`**.
5. Specificate i registri di distribuzione degli CodeDeploy agenti, come segue:
   1. Per`Log file path`, inserire il percorso del file di registro CodeDeploy di distribuzione:**`C:\ProgramData\Amazon\CodeDeploy\deployment-logs\codedeploy-agent-deployments.log`**.

2. Per `Log group name` inserire il nome di un gruppo di log, ad esempio:**`codedeploy-agent-deployment-log`**.
3. Per `Log stream name` inserire il nome di un flusso di log, per esempio:**`{instance_id}-codedeploy-agent-deployment-log`**.

Per configurare l' CloudWatch agente creando o modificando manualmente un file di configurazione (Linux)

1. Creare o modificare il file di configurazione dell' CloudWatch agente come descritto in Creare o modificare manualmente il file di configurazione dell' CloudWatch agente.
2. Assicuratevi che il file venga chiamato `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` e che contenga il codice seguente:

   Il percorso del registro dell'aggiornamento varia in base alla versione. Per la versione 2.0.x e successive, utilizzare. `/var/log/aws/codedeploy-agent/codedeploy-agent-updater.log` Per la versione 1.8.x e precedenti, usa. `/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"
                }
            ]
        }
    }
}
...
```

Per configurare l' CloudWatch agente creando o modificando manualmente un file di configurazione (Windows)

1. Creare o modificare il file di configurazione dell' CloudWatch agente come descritto in Creare o modificare manualmente il file di configurazione dell' CloudWatch agente.
2. Assicuratevi che il file venga chiamato `C:\ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json` e che contenga il codice seguente:

   Per la versione 2.0.x e successive, il nome del file di registro dell'agente è. `codedeploy-agent.log` Per la versione 1.8.x e precedenti, il nome del file di log dell'agente è. `codedeploy-agent-log.txt` Il percorso del registro di distribuzione è lo stesso per entrambe le versioni.

   ```
   ...
   "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"
                       }
                   ]
               },
   ```

```
            ...
        }
    },
...
```

Riavviare l' CloudWatch agente

Dopo aver apportato le modifiche, riavvia l' CloudWatch agente come descritto in Avvio dell' CloudWatch agente.

Utilizzo delle istanze per CodeDeploy

CodeDeploy supporta distribuzioni su istanze che eseguono Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) e Windows Server.

Puoi utilizzarlo CodeDeploy per eseguire la distribuzione sia su istanze Amazon EC2 che su istanze locali. Un'istanza locale è qualsiasi dispositivo fisico diverso da un'istanza Amazon EC2 in grado di eseguire CodeDeploy l'agente e connettersi agli endpoint del AWS servizio pubblico. Puoi utilizzarla CodeDeploy per distribuire contemporaneamente un'applicazione su istanze Amazon EC2 nel cloud e su PC desktop in ufficio o server nel tuo data center.

Confronto tra istanze Amazon EC2 e istanze locali

La tabella seguente mette a confronto le istanze di Amazon EC2 e le istanze locali:

Oggetto	Istanze Amazon EC2	On-premises Istanze
Richiede l'installazione e l'esecuzione di una versione dell' CodeDeploy agente compatibile con il sistema operativo in esecuzione sull'istanza.	Sì	Sì
Richiede che l'istanza sia in grado di connettersi a CodeDeploy.	Sì	Sì
Richiede il collegamento di un profilo di istanza IAM all'istanza. Il profilo dell'istanza IAM deve disporre delle autorizzazioni per partecipare alle CodeDeploy distribuzioni. Per informazioni, consulta Fase 4: crea un profilo di istanza	Sì	No

Oggetto	Istanze Amazon EC2	On-premises Istanze
IAM per le tue istanze Amazon EC2.		
Richiede che vengano eseguite le operazioni elencate di seguito per autenticare e registrare le istanze: • Crea un ruolo IAM che può essere assunto da un utente IAM su ogni istanza per recuperare credenziali temporanee periodicamente aggiornate tramite. Servizio di token di sicurezza AWS • Crea un utente IAM per ogni istanza e archivia le credenziali dell'account dell'utente IAM in testo semplice sull'istanza.	No	Sì
Richiede la registrazione di ogni istanza CodeDeploy prima di poterla implementare.	No	Sì
Richiede di etichettare ogni istanza prima di CodeDeploy poter essere distribuita su di essa.	Sì	Sì
Può partecipare agli scenari di Amazon EC2 Auto Scaling ed Elastic Load Balancing come parte delle distribuzioni. CodeDeploy	Sì	No

Oggetto	Istanze Amazon EC2	On-premises Istanze
Può essere distribuito da GitHub bucket e repository Amazon S3.	Sì	Sì
Può supportare i trigger che richiedono l'invio di notifiche SMS o e-mail quando nelle distribuzioni o nelle istanze si verificano eventi specifici.	Sì	Sì
È soggetta a essere fatturata per le distribuzioni associate.	No	Sì

Attività di istanza per CodeDeploy

Per avviare o configurare le istanze per l'uso nelle distribuzioni, scegliere tra le seguenti istruzioni:

Voglio lanciare una nuova istanza Amazon Linux o Windows Server Amazon EC2.	Per avviare l'istanza Amazon EC2 con il minimo sforzo, consulta. Crea un'istanza Amazon EC2 per (CodeDeploy CloudFormation modello) Per avviare l'istanza Amazon EC2 principal mente da soli, consulta. Crea un'istanza Amazon EC2 per (CodeDeploy AWS CLI o console Amazon EC2)
Voglio lanciare una nuova istanza di Ubuntu Server o RHEL Amazon EC2.	Per informazioni, consulta Crea un'istanza Amazon EC2 per (CodeDeploy AWS CLI o console Amazon EC2).
Voglio configurare un'istanza Amazon Linux, Windows Server, Ubuntu Server o RHEL Amazon EC2.	Per informazioni, consulta Configura un'istanza Amazon EC2 con cui lavorare CodeDeploy.

Voglio configurare un'istanza locale di Windows Server, Ubuntu Server o RHEL (dispositivi fisici che non sono istanze Amazon EC2).	Per informazioni, consulta Working with On-Premises Instances.
Desidero effettuare CodeDeploy il provision ing di un parco istanze sostitutivo durante una distribuzione. blue/green	Per informazioni, consulta Utilizzo delle distribuzioni in CodeDeploy.

Per preparare le istanze Amazon EC2 nei gruppi Amazon EC2 Auto Scaling, devi seguire alcuni passaggi aggiuntivi. Per ulteriori informazioni, consulta Integrazione CodeDeploy con Amazon EC2 Auto Scaling.

Argomenti

- Tagging Instances for Deployments
- Working with Amazon EC2 Instances
- Working with On-Premises Instances
- View Instance Details
- Instance Health

Taggare le istanze per i gruppi di distribuzione in CodeDeploy

Per aiutarti a gestire le istanze Amazon EC2 e le istanze locali, puoi utilizzare i tag per assegnare i tuoi metadati a ciascuna risorsa. I tag ti consentono di categorizzare le tue istanze in modi diversi (ad esempio, per scopo, proprietario o ambiente). Ciò risulta particolarmente utile quando si dispone di numerose istanze. È possibile individuare rapidamente un'istanza o un gruppo di istanze in base al tag assegnato. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. Per ulteriori informazioni, consulta Tagging your Amazon EC2 resources.

Per specificare quali istanze sono incluse in un gruppo di CodeDeploy distribuzione, devi specificare i tag in uno o più gruppi di tag. Le istanze che soddisfano i criteri di tag sono quelle su cui è installata l'ultima versione dell'applicazione quando viene creata una distribuzione nel gruppo di distribuzione.

> **Note**
>
> Puoi anche includere gruppi di Amazon EC2 Auto Scaling nei gruppi di distribuzione, ma sono identificati dai loro nomi anziché dai tag applicati alle istanze. Per informazioni, consulta Integrazione CodeDeploy con Amazon EC2 Auto Scaling.

I criteri per le istanze in un gruppo di distribuzione possono essere semplici come un unico tag in un singolo gruppo di tag. È possibile includere fino a 10 tag in totale in un massimo di tre gruppi di tag.

Se si utilizza un singolo gruppo di tag, qualsiasi istanza del gruppo identificata da almeno un tag viene inclusa nel gruppo di distribuzione. Se si utilizzano più gruppi di tag, vengono incluse solo le istanze che sono identificate da almeno un tag in ciascuno dei gruppi di tag.

I seguenti esempi illustrano come possono essere utilizzati i tag e i gruppi di tag per selezionare le istanze per un gruppo di distribuzione.

Argomenti

- Esempio 1: gruppo di tag singolo, tag singolo
- Esempio 2: gruppo di tag singolo, tag multipli
- Esempio 3: gruppi di tag multipli, tag singoli
- Esempio 4: più gruppi di tag, più tag

Esempio 1: gruppo di tag singolo, tag singolo

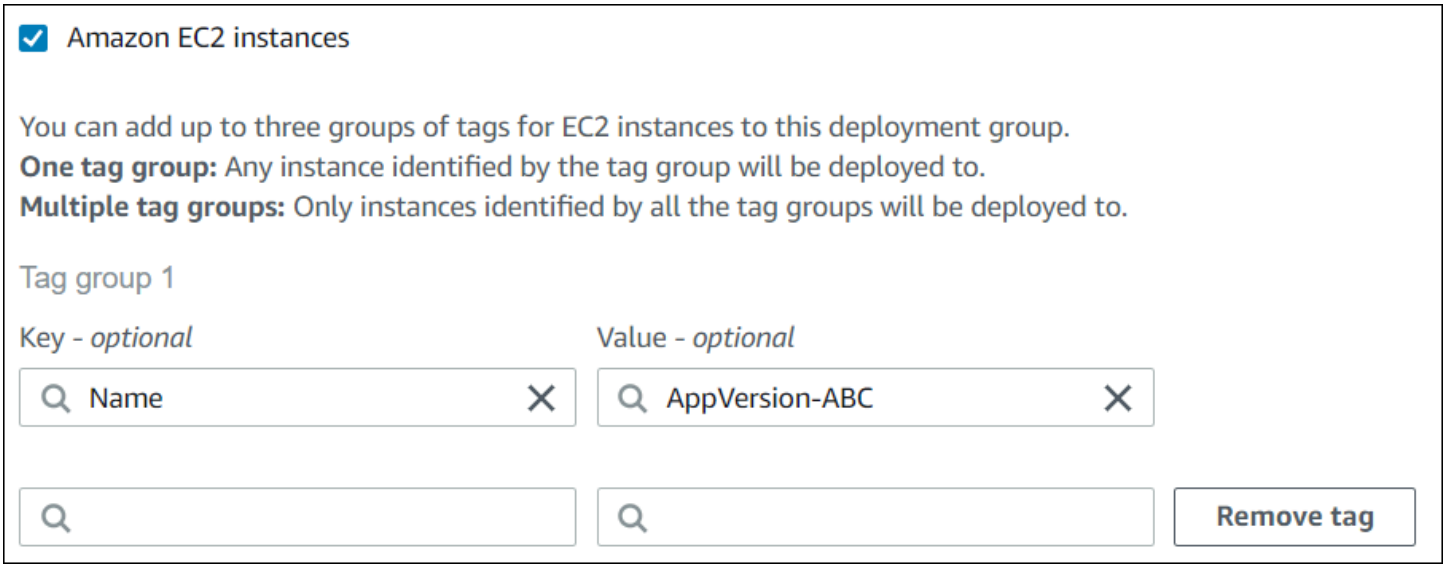
È possibile specificare un singolo tag in un singolo gruppo di tag:

Gruppo di tag 1

Chiave	Valore
Nome	AppVersion-ABC

Ogni istanza che contiene il tag `Name=AppVersion-ABC` è parte del gruppo di distribuzione, anche a essa sono applicati altri tag.

CodeDeploy visualizzazione della configurazione della console:



Struttura JSON:

```
"ec2TagSet": {
         "ec2TagSetList": [
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Name",
                  "Value": "AppVersion-ABC"
               }
            ]
         ]
      },
```

Esempio 2: gruppo di tag singolo, tag multipli

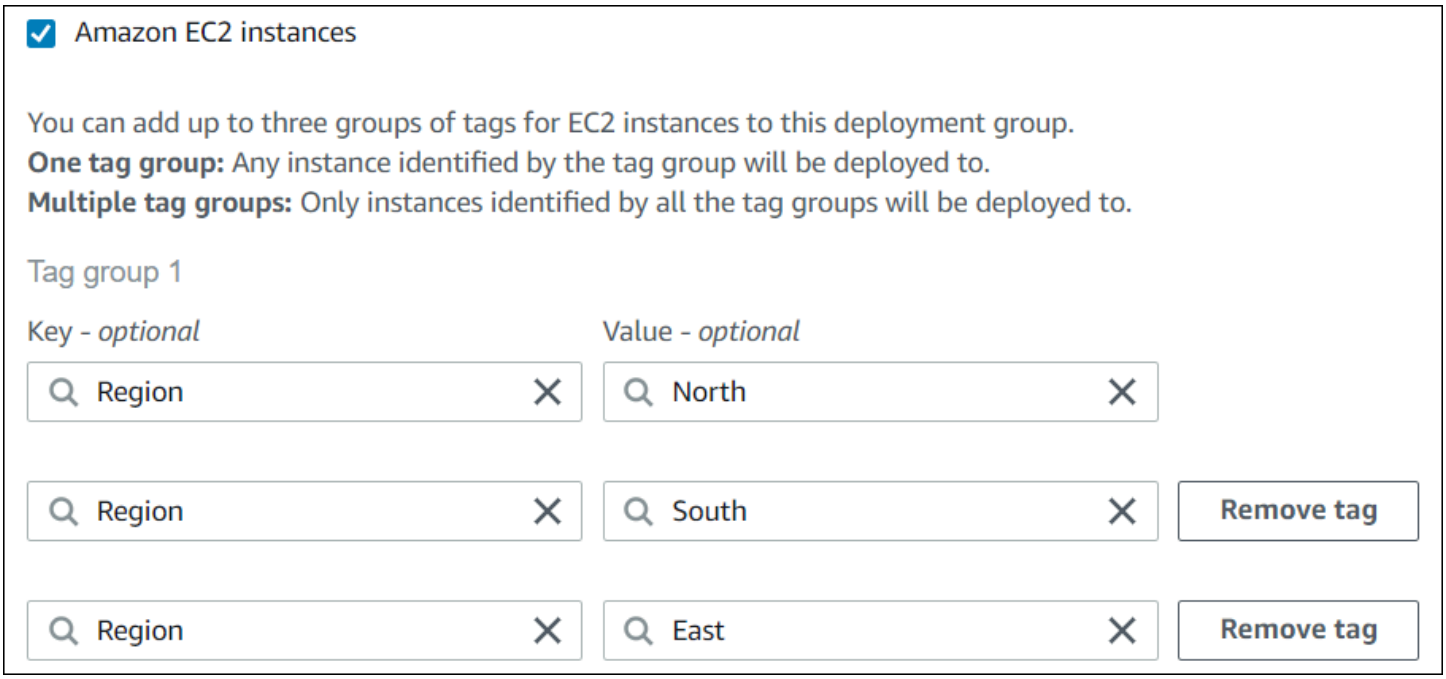
È possibile anche specificare tag multipli in un singolo gruppo di tag:

Gruppo di tag 1

Chiave	Valore
Region	Nord
Region	Sud
Region	Est

Un'istanza che contiene il tag con uno qualsiasi di questi tre tag è parte del gruppo di distribuzione, anche se presenta altri tag applicati. Se, ad esempio, si dispone di altre istanze con il tag `Region=West`, queste potrebbero non essere incluse nel gruppo di distribuzione.

CodeDeploy visualizzazione della configurazione della console:



Struttura JSON:

```
"ec2TagSet": {
         "ec2TagSetList": [
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "North"
               },
                           {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "South"
               },
                           {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "East"
               }
```

```
            ]
         ]
      },
```

Esempio 3: gruppi di tag multipli, tag singoli

È inoltre possibile utilizzare più insiemi di gruppi di tag con una singola coppia chiave-valore in ciascuno per specificare i criteri per le istanze in un gruppo di distribuzione. Quando si utilizzano più gruppi di tag in un gruppo di distribuzione, sono incluse nel gruppo di distribuzione solo le istanze che sono identificate da tutti i gruppi di tag.

Gruppo di tag 1

Chiave	Valore
Nome	AppVersion-ABC

Gruppo di tag 2

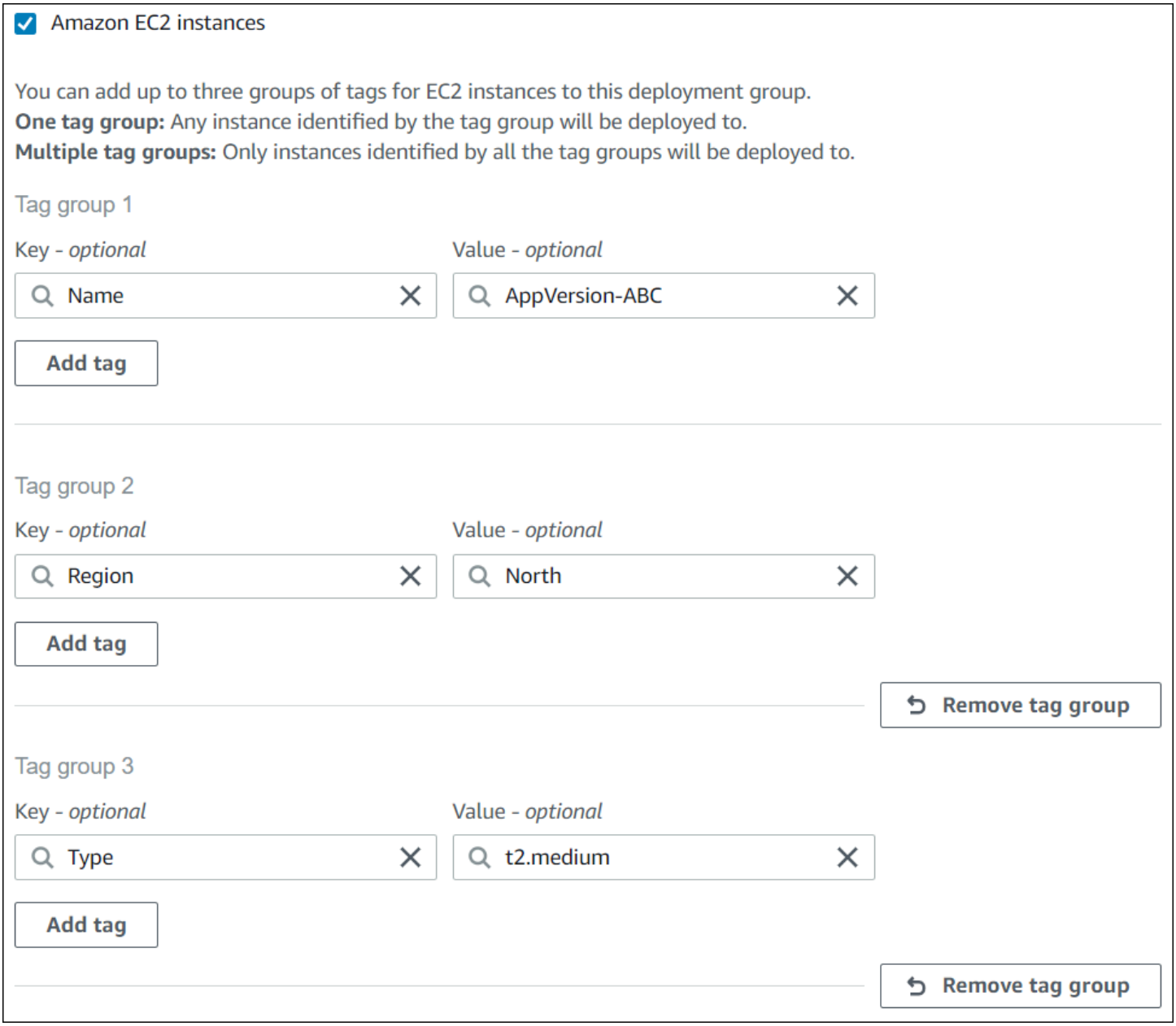
Chiave	Valore
Region	Nord

Gruppo di tag 3

Chiave	Valore
Tipo	t2.medium

Potresti avere istanze in molte regioni e di vari tipi di istanze etichettate con`Name=AppVersion-ABC`. In questo esempio, solo le istanze che sono contrassegnate anche con i tag `Region=North` e `Type=t2.medium` fanno parte del gruppo di distribuzione.

CodeDeploy vista di configurazione della console:



Struttura JSON:

```
"ec2TagSet": {
         "ec2TagSetList": [
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Name",
                  "Value": "AppVersion-ABC"
               }
            ],
```

```
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "North"
               }
            ],
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Type",
                  "Value": "t2.medium"
               }
            ],
         ]
      },
```

Esempio 4: più gruppi di tag, più tag

Quando usi più gruppi di tag con diversi tag in uno o più gruppi, un'istanza deve corrispondere ad almeno uno dei tag in ciascuno dei gruppi.

Gruppo di tag 1

Chiave	Valore
Ambiente	Beta
Ambiente	Gestione temporanea

Gruppo di tag 2

Chiave	Valore
Region	Nord
Region	Sud
Region	Est

Gruppo di tag 3

Chiave	Valore
Tipo	t2.medium
Tipo	t2.large

In questo esempio, per essere inclusa nel gruppo di distribuzione, un'istanza deve avere un tag (1) `Environment=Beta` o `Environment=Staging`, con (2) `Region=North`, `Region=South` oppure `Region=East` e con (3) `Type=t2.medium` o `Type=t2.large`.

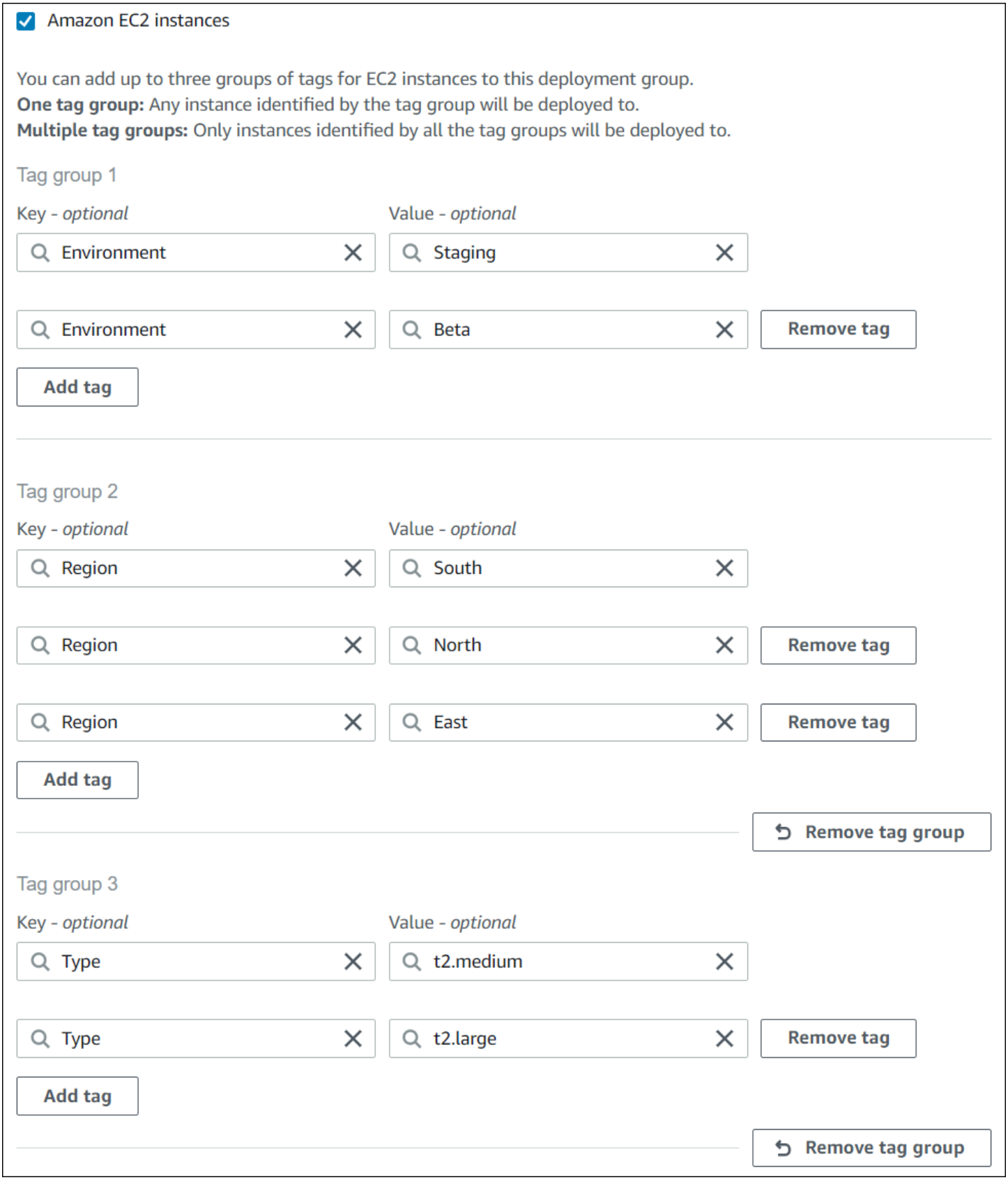
A scopo illustrativo, le istanze con i seguenti gruppi di tag sarebbero tra quelle incluse nel gruppo di distribuzione:

- `Environment=Beta`, `Region=North`,`Type=t2.medium`
- `Environment=Staging`,`Region=East`,`Type=t2.large`
- `Environment=Staging`,`Region=South`,`Type=t2.large`

Le istanze con i seguenti gruppi di tag non sarebbero tra quelle incluse nel gruppo di distribuzione. I valori chiave evidenziati causano l'esclusione delle istanze:

- `Environment=Beta`, Region=West,`Type=t2.medium`
- `Environment=Staging`,`Region=East`,Type=t2.micro
- Environment=Production,`Region=South`,`Type=t2.large`

CodeDeploy visualizzazione della configurazione della console:


Amazon EC2 instances
You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.
Tag group 1
Key - optional
Value - optional
Environment
Staging
Environment
Beta
Remove tag
Add tag
Tag group 2
Key - optional
Value - optional
Region
South
Region
North
Remove tag
Region
East
Remove tag
Add tag
Remove tag group
Tag group 3
Key - optional
Value - optional
Type
t2.medium
Type
t2.large
Remove tag
Add tag
Remove tag group

Struttura JSON:

```
"ec2TagSet": {
         "ec2TagSetList": [
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Environment",
                  "Value": "Beta"
               },
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Environment",
                  "Value": "Staging"
               }
            ],
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "North"
               },
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "South"
               },
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "East"
               }
            ],
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Type",
                  "Value": "t2.medium"
               },
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Type",
                  "Value": "t2.large"
               }
```

```
            ],
         ]
      },
```

Utilizzo delle istanze Amazon EC2 per CodeDeploy

Un'istanza Amazon EC2 è un ambiente di elaborazione virtuale che puoi creare e configurare utilizzando Amazon Elastic Compute Cloud. Amazon EC2 offre capacità di elaborazione scalabile nel cloud. AWS Puoi usare Amazon EC2 per lanciare tutti o pochi server virtuali di cui hai bisogno per le tue CodeDeploy distribuzioni.

Per ulteriori informazioni su Amazon EC2, consulta la Amazon EC2 Getting Started Guide.

Le istruzioni in questa sezione mostrano come creare e configurare istanze Amazon EC2 da utilizzare nelle distribuzioni. CodeDeploy

Argomenti

- Crea un'istanza Amazon EC2 per (CodeDeploy AWS CLI o console Amazon EC2)
- Crea un'istanza Amazon EC2 per (CodeDeploy CloudFormation modello)
- Configura un'istanza Amazon EC2 con cui lavorare CodeDeploy

Crea un'istanza Amazon EC2 per (CodeDeploy AWS CLI o console Amazon EC2)

Queste istruzioni mostrano come avviare una nuova istanza Amazon EC2 configurata per l'uso nelle CodeDeploy distribuzioni.

Puoi utilizzare il nostro CloudFormation modello per avviare un'istanza Amazon EC2 che esegue Amazon Linux o Windows Server già configurata per l'uso nelle CodeDeploy distribuzioni. Non forniamo un CloudFormation modello per le istanze Amazon EC2 che eseguono Ubuntu Server o Red Hat Enterprise Linux (RHEL). Per le alternative per l'uso del modello, consulta Utilizzo delle istanze per CodeDeploy.

Puoi utilizzare la console Amazon EC2 o le API di Amazon EC2 per avviare un'istanza Amazon EC2. AWS CLI

Avvia un'istanza Amazon EC2 (console)

Prerequisiti

Se non l'hai già fatto, segui le istruzioni riportate Guida introduttiva con CodeDeploy per configurare AWS CLI e creare un profilo di istanza IAM.

Avviare un'istanza Amazon EC2

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Nel riquadro di navigazione, scegliere Instances (Istanze), quindi selezionare Launch Instance (Avvia istanza).
3. Nella pagina Fase 1: scegliere una Amazon Machine Image (AMI),, dalla scheda Quick Start, individuare il sistema operativo e la versione che si desidera utilizzare e quindi scegliere Seleziona. Devi scegliere un sistema operativo AMI Amazon EC2 supportato da. CodeDeploy Per ulteriori informazioni, consulta Sistemi operativi supportati dall'agente CodeDeploy.
4. Nella pagina Fase 2: Scegli un tipo di istanza, scegli qualsiasi tipo di istanza Amazon EC2 disponibile, quindi scegli Avanti: Configure Instance Details.
5. Nella pagina Fase 3: Configurazione dei dettagli dell'istanza, nell'elenco dei ruoli IAM, scegli il ruolo dell'istanza IAM in Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2 cui hai creato. Se hai utilizzato il nome del ruolo suggerito, scegli **`CodeDeployDemo-EC2-Instance-Profile`**. Se hai creato il nome del ruolo, sceglilo.

   > **Note**
   >
   > Se un cloud privato virtuale (VPC) predefinito non viene visualizzato nell'elenco Rete, devi scegliere o creare un Amazon VPC e una sottorete. Scegli Crea nuovo VPC o Crea una nuova sottorete o entrambe le opzioni. Per ulteriori informazioni, consulta VPC e sottoreti.

6. Scegli Passaggio successivo: aggiunta dello storage.
7. Lascia la pagina Step 4: Add Storage (Fase 4: aggiunta spazio di archiviazione) invariata e seleziona Next: Add Tags (Fase successiva: aggiunta di tag).
8. Nella pagina Step 5: Add Tags (Fase 5: aggiunta di tag), scegli Add tag (Aggiungi tag).
9. Nella casella Key (Chiave), digita **`Name`**. Nella casella Value (Valore), digita **`CodeDeployDemo`**.

> **Important**
>
> Per il contenuto delle caselle Key (Chiave) e Value (Valore) viene rilevata la distinzione tra maiuscole e minuscole.

10. Scegliere Next: Configure Security Group (Fase successiva: configurazione del gruppo di sicurezza).
11. Nella pagina Step 6: Configure Security Group (Fase 6: configurare i gruppi di sicurezza) verificare che l'opzione Create a new security group (Crea un nuovo gruppo di sicurezza) sia selezionata.

    Un ruolo SSH predefinito è configurato per le istanze Amazon EC2 che eseguono Amazon Linux, Ubuntu Server o RHEL. Un ruolo RDP predefinito è configurato per le istanze Amazon EC2 che eseguono Windows Server.
12. Per aprire la porta HTTP, scegliere il pulsante Add Rule (Aggiungi regola) quindi, dall'elenco a discesa Tipo, scegliere **HTTP**. Accetta il valore Source predefinito di Custom 0.0.0. 0/0, quindi scegli Review and Launch.

    > **Note**
    >
    > In un ambiente di produzione, consigliamo di limitare l'accesso alle porte SSH, RDP e HTTP, invece di specificare Anywhere 0.0.0. 0/0. CodeDeploy non richiede l'accesso illimitato alle porte e non richiede l'accesso HTTP. Per ulteriori informazioni, consulta Suggerimenti per proteggere l'istanza Amazon EC2.

    Se viene visualizzata la finestra di dialogo Boot from General Purpose (SSD) (Avvio da General Purpose (SSD)), seguire le istruzioni e scegliere Next (Avanti).
13. Lasciare invariate le impostazioni nella pagina Step 7: Review Instance Launch (Fase 7: verificare l'avvio dell'istanza) e scegliere Launch (Avvia).
14. Nella finestra di dialogo Select an existing key pair or create a new key pair (Seleziona una coppia di chiavi esistenti o crea una nuova coppia di chiavi) scegliere Choose an existing key pair (Scegli una coppia di chiavi esistente) o Create a new key pair (Crea una nuova coppia di chiavi). Se hai già configurato una coppia di chiavi dell'istanza Amazon EC2 instance, è possibile sceglierla in questa pagina.

Se invece non hai ancora una coppia di chiavi dell'istanza Amazon EC2, scegli Create a new key pair (Crea una nuova coppia di chiavi) e assegnale un nome riconoscibile. Scegli Scarica Key Pair per scaricare la coppia di chiavi dell'istanza Amazon EC2 sul tuo computer.

> **Important**
>
> È necessario disporre di una coppia di chiavi se si desidera accedere all'istanza Amazon EC2 con SSH o RDP.

15. Scegliere Launch Instances (Avvia istanze).
16. Scegli l'ID per la tua istanza Amazon EC2. Non continuare fino a quando l'istanza non viene avviata e non ha superato tutti i controlli.

Installa l'agente CodeDeploy

L' CodeDeploy agente deve essere installato sull'istanza Amazon EC2 prima di utilizzarlo nelle CodeDeploy distribuzioni. Per ulteriori informazioni, consulta Installa l' CodeDeploy agente.

> **Note**
>
> Puoi configurare l'installazione e gli aggiornamenti automatici dell' CodeDeploy agente quando crei il gruppo di distribuzione nella console.

Avvia un'istanza Amazon EC2 (CLI)

Prerequisiti

Se non l'hai già fatto, segui le istruzioni riportate Guida introduttiva con CodeDeploy per impostare e configurare AWS CLI e creare un profilo di istanza IAM.

Avviare un'istanza Amazon EC2

1. Solo per Windows Server Se stai creando un'istanza Amazon EC2 che esegue Windows Server, chiama i authorize-security-group-ingress comandi create-security-group and per creare un gruppo di sicurezza che consenta l'accesso RDP (che non è consentito di default) e, in alternativa, l'accesso HTTP. Ad esempio, per creare un gruppo di sicurezza denominato CodeDeployDemo-Windows-Security-Group, eseguire i comandi seguenti, uno alla volta:

```
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**
>
> A scopo dimostrativo, questi comandi creano un gruppo di sicurezza che consente l'accesso illimitato per RDP attraverso la porta 3389 e, in alternativa, l'accesso HTTP attraverso la porta 80. Come best practice, è consigliabile limitare l'accesso alle porte RDP e HTTP. CodeDeploy non richiede l'accesso illimitato alle porte e non richiede l'accesso HTTP. Per ulteriori informazioni, consulta Suggerimenti per proteggere l'istanza Amazon EC2.

2. Chiama il run-instances comando per creare e avviare l'istanza Amazon EC2.

   Prima di chiamare questo comando, devi raccogliere le informazioni seguenti:

   - L'ID di un'Amazon Machine Image (AMI) (*ami-id*) che usi per l'istanza. Per ottenere l'ID, consulta Ricerca di un'AMI adatta.
   - Il nome del tipo di istanza Amazon EC2 (*instance-type*) che crei, ad esempio. `t1.micro` Per un elenco, consulta Tipi di istanze Amazon EC2.
   - Il nome di un profilo di istanza IAM con autorizzazione ad accedere al bucket Amazon S3 in cui sono archiviati i file di installazione dell' CodeDeploy agente per la tua regione.

     Per informazioni sulla creazione di un profilo di istanza IAM, consulta. Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2
   - Il nome di una coppia di chiavi di istanza Amazon EC2 (*key-name*) per abilitare l'accesso SSH a un'istanza Amazon EC2 che esegue Amazon Linux, Ubuntu Server o l'accesso RHEL o RDP a un'istanza Amazon EC2 che esegue Windows Server.

> **Important**
>
> Digita solo il nome della coppia di chiavi, non l'estensione del file della coppia di chiavi. Ad esempio, my-keypair, non my-keypair.pem.

Per trovare il nome di una coppia di key pair, apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2 Nel pannello di navigazione, in Network & Security (Rete e sicurezza), scegliere Key Pairs (Coppie di chiavi) e annotare il nome della coppia di chiavi nell'elenco.

Per generare una coppia di chiavi, consulta Creazione di una coppia di chiavi con Amazon EC2. Assicurati di creare la key pair in una delle regioni elencate in Regione e gli endpoint in Riferimenti generali di AWS. Altrimenti, non sarai in grado di utilizzare la coppia di chiavi dell'istanza Amazon EC2 con. CodeDeploy

Per Amazon Linux, RHEL e Ubuntu Server

Per richiamare il run-instances comando per avviare un'istanza Amazon EC2 che esegue Amazon Linux, Ubuntu Server o RHEL e collegare il profilo dell'istanza IAM in cui è stato creato. Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2 Ad esempio:

```
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**
>
> Questo comando crea un gruppo di sicurezza predefinito per l'istanza Amazon EC2 che consente l'accesso a diverse porte, incluso l'accesso illimitato per SSH tramite la porta 22 e, in alternativa, HTTP tramite la porta 80. Come best practice, consigliamo di limitare l'accesso solo alle porte SSH e HTTP. CodeDeploy non richiede l'accesso illimitato

alle porte e non richiede l'accesso alla porta HTTP. Per ulteriori informazioni, consulta Suggerimenti per proteggere l'istanza Amazon EC2.

Per Windows Server

Richiamare il run-instances comando per avviare un'istanza Amazon EC2 che esegue Windows Server e collegare il profilo dell'istanza IAM in Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2 cui è stato creato e specificare il nome del gruppo di sicurezza creato nel passaggio 1. Ad esempio:

```
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
```

Questi comandi avviano una singola istanza Amazon EC2 con l'AMI, la coppia di chiavi e il tipo di istanza specificati, con il profilo di istanza IAM specificato ed eseguono lo script specificato durante l'avvio.

3. Prendere nota del valore di `InstanceID` nell'output. Se dimentichi questo valore, puoi ottenerlo in un secondo momento chiamando il describe-instances comando sulla coppia di chiavi dell'istanza Amazon EC2.

```
aws ec2 describe-instances --filters "Name=key-name,Values=keyName" --query
 "Reservations[*].Instances[*].[InstanceId]" --output text
```

Usa l'ID dell'istanza per chiamare il create-tags comando, che contrassegna l'istanza Amazon EC2 in modo che CodeDeploy possa trovarla in un secondo momento durante una distribuzione. Nell'esempio seguente, il tag è denominato**`CodeDeployDemo`**, ma puoi specificare qualsiasi tag di istanza Amazon EC2 che desideri.

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=CodeDeployDemo
```

È possibile applicare più tag alla stessa istanza contemporaneamente. Ad esempio:

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=testInstance
 Key=Region,Value=West Key=Environment,Value=Beta
```

Per verificare che l'istanza Amazon EC2 sia stata avviata e abbia superato tutti i controlli, usa l'ID dell'istanza per chiamare il describe-instance-status comando.

```
aws ec2 describe-instance-status --instance-ids instance-id --query
 "InstanceStatuses[*].InstanceStatus.[Status]" --output text
```

Se l'istanza è stata avviata e ha superato tutti i controlli, nell'output sarà presente `ok`:

Installa l'agente CodeDeploy

L' CodeDeploy agente deve essere installato sull'istanza Amazon EC2 prima di utilizzarlo nelle CodeDeploy distribuzioni. Per ulteriori informazioni, consulta Installa l' CodeDeploy agente.

> **Note**
>
> Puoi configurare l'installazione e gli aggiornamenti automatici dell' CodeDeploy agente quando crei il gruppo di distribuzione nella console.

Crea un'istanza Amazon EC2 per (CodeDeploy CloudFormation modello)

Puoi utilizzare il nostro CloudFormation modello per avviare rapidamente un'istanza Amazon EC2 che esegue Amazon Linux o Windows Server. Puoi utilizzare la AWS CLI, la CodeDeploy console o le AWS API per avviare l'istanza con il modello. Oltre a lanciare l'istanza, il modello effettua le seguenti operazioni:

- Indica di CloudFormation concedere all'istanza l'autorizzazione a partecipare CodeDeploy alle distribuzioni.
- Etichetta l'istanza in modo da CodeDeploy poterla trovare durante una distribuzione.
- Installa ed esegue l' CodeDeploy agente sull'istanza.

Non è necessario utilizzare la nostra CloudFormation per configurare un'istanza Amazon EC2. Per le alternative, consulta Utilizzo delle istanze per CodeDeploy.

Non forniamo un CloudFormation modello per le istanze Amazon EC2 che eseguono Ubuntu Server o Red Hat Enterprise Linux (RHEL).

Argomenti

- Prima di iniziare
- Avvia un'istanza Amazon EC2 con CloudFormation modello (console)
- Avvia un'istanza Amazon EC2 con CloudFormation modello (AWS CLI)

Prima di iniziare

Prima di poter utilizzare il CloudFormation modello per avviare le istanze Amazon EC2, assicurati di completare i seguenti passaggi.

1. Assicurati di aver creato un utente amministrativo, come descritto in. Fase 1: Configurazione Double-check che l'utente disponga delle seguenti autorizzazioni minime e aggiungi quelle non presenti:

   - formazione di nuvole: *
   - codedeploy:*
   - ec2:*
   - obiettivo: AddRoleToInstanceProfile
   - sono: CreateInstanceProfile
   - sono: CreateRole
   - sono: DeleteInstanceProfile
   - sono: DeleteRole
   - sono: DeleteRolePolicy
   - sono: GetRole
   - sono: DeleteRolePolicy
   - sono: PutRolePolicy
   - sono: RemoveRoleFromInstanceProfile
2. Assicurati di disporre di una coppia di chiavi di istanza per abilitare l'accesso SSH all'istanza Amazon EC2 che esegue Amazon Linux o l'accesso RDP all'istanza che esegue Windows Server.

   Per trovare il nome di una coppia di key pair, apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2 Nel pannello di navigazione, in Network & Security (Rete e

sicurezza), scegliere Key Pairs (Coppie di chiavi) e annotare il nome della coppia di chiavi nell'elenco.

Per generare una nuova coppia di chiavi, consulta Creating your key pair using Amazon EC2. Assicurati che la key pair sia creata in una delle regioni elencate in Regione e che gli endpoint in Riferimenti generali di AWS. Altrimenti, non sarà possibile utilizzare la coppia di chiavi con CodeDeploy.

Avvia un'istanza Amazon EC2 con CloudFormation modello (console)

1. Accedi a Console di gestione AWS e apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.

   > **Important**
   >
   > Accedi Console di gestione AWS con lo stesso account che hai utilizzatoGuida introduttiva con CodeDeploy. Nella barra di navigazione, nel selettore Regione, scegli una delle regioni elencate in Regione e terminali in. Riferimenti generali di AWS CodeDeploy supporta solo queste regioni.

2. Scegli Crea stack.
3. In Scegli un modello, scegli Specificare l'URL di un modello Amazon S3. Nella casella, digita la posizione del CloudFormation modello per la tua regione, quindi scegli Avanti.

Region	Ubicazione del CloudFormation modello
Stati Uniti orientali (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Stati Uniti orientali (Virginia settentrionale)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Ubicazione del CloudFormation modello
Regione Stati Uniti occidentali (California settentrionale)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Stati Uniti occidentali (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Canada (Centrale)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Europa (Irlanda)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Europa (Londra)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Europa (Parigi)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Europa (Francoforte)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Ubicazione del CloudFormation modello
Regione di Israele (Tel Aviv)	`http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Asia Pacifico (Hong Kong)	`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`
Regione Asia Pacifico (Seoul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Asia Pacifico (Singapore)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Asia Pacifico (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Ubicazione del CloudFormation modello
Regione Asia Pacifico (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Asia Pacifico (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Sud America (San Paolo)	`aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`

4. Nella casella Stack name (Nome stack) digitare il nome dello stack (ad esempio **`CodeDeployDemoStack`**).
5. In Parameters (Parametri), digita quanto segue e scegli Avanti

   - Per InstanceCount, digita il numero di istanze che desideri avviare. (Ti consigliamo di lasciare l'impostazione predefinita di 1).
   - Ad esempio InstanceType, digita il tipo di istanza che desideri avviare (o lascia il valore predefinito di t1.micro).
   - Per KeyPairName, digitare il nome della coppia di chiavi dell'istanza. Digita solo il nome della coppia di chiavi, non l'estensione del file della coppia di chiavi.
   - Per OperatingSystembox, digita **`Windows`** per avviare le istanze che eseguono Windows Server (o lascia l'impostazione predefinita di Linux).
   - Per SSHLocation, digita l'intervallo di indirizzi IP da utilizzare per la connessione all'istanza con SSH o RDP (oppure lascia il valore predefinito 0.0.0). 0/0).

> **Important**
>
> L'impostazione predefinita di **0.0.0.0/0** viene fornita solo a scopo dimostrativo. CodeDeploy non richiede che le istanze Amazon EC2 abbiano accesso illimitato alle porte. Come best practice, è consigliabile limitare l'accesso a porte SSH (e HTTP). Per ulteriori informazioni, consulta Suggerimenti per proteggere l'istanza Amazon EC2.

- Ad esempio TagKey, digita la chiave del tag dell'istanza CodeDeploy che utilizzerà per identificare le istanze durante la distribuzione (oppure lascia il valore predefinito di Name).
- Ad esempio TagValue, digita il valore del tag dell'istanza che CodeDeploy utilizzerà per identificare le istanze durante la distribuzione (o lascia il valore predefinito di CodeDeployDemo).

6. Nella pagina Opzioni, lasciare le caselle opzione vuote e scegliere Successivo.

> **Important**
>
> CloudFormation i tag sono diversi dai CodeDeploy tag. CloudFormation utilizza i tag per semplificare l'amministrazione dell'infrastruttura. CodeDeploy utilizza tag per identificare le istanze Amazon EC2. I tag CodeDeploy specificati nella pagina Specificare i parametri.

7. Nella pagina di revisione, in Capacità, seleziona la casella Riconosco che CloudFormation potrebbe creare risorse IAM, quindi scegli Crea.

   Dopo CloudFormation aver creato lo stack e avviato le istanze Amazon EC2, nella CloudFormation console, CREATE_COMPLETE verrà visualizzato nella colonna Stato. Questo processo può richiedere alcuni minuti.

Per verificare che l' CodeDeploy agente sia in esecuzione sulle istanze Amazon EC2Gestione delle operazioni degli CodeDeploy agenti, consulta e procedi con. Crea un'applicazione con CodeDeploy

Avvia un'istanza Amazon EC2 con CloudFormation modello (AWS CLI)

1. Usa il nostro CloudFormation modello in una chiamata al create-stack comando. Questo stack lancerà una nuova istanza Amazon EC2 con CodeDeploy l'agente installato.

   Per avviare un'istanza Amazon EC2 che esegue Amazon Linux:

```
aws cloudformation create-stack \
  --stack-name CodeDeployDemoStack \
  --template-url templateURL \
  --parameters ParameterKey=InstanceCount,ParameterValue=1
 ParameterKey=InstanceType,ParameterValue=t1.micro \
    ParameterKey=KeyPairName,ParameterValue=keyName
 ParameterKey=OperatingSystem,ParameterValue=Linux \
    ParameterKey=SSHLocation,ParameterValue=0.0.0.0/0
 ParameterKey=TagKey,ParameterValue=Name \
    ParameterKey=TagValue,ParameterValue=CodeDeployDemo \
  --capabilities CAPABILITY_IAM
```

Per avviare un'istanza Amazon EC2 che esegue Windows Server:

```
aws cloudformation create-stack --stack-name CodeDeployDemoStack --template-
url template-url --parameters ParameterKey=InstanceCount,ParameterValue=1
 ParameterKey=InstanceType,ParameterValue=t1.micro
 ParameterKey=KeyPairName,ParameterValue=keyName
 ParameterKey=OperatingSystem,ParameterValue=Windows
 ParameterKey=SSHLocation,ParameterValue=0.0.0.0/0
 ParameterKey=TagKey,ParameterValue=Name
 ParameterKey=TagValue,ParameterValue=CodeDeployDemo --capabilities CAPABILITY_IAM
```

*keyName*è il nome della key pair dell'istanza. Digita solo il nome della coppia di chiavi, non l'estensione del file della coppia di chiavi.

*template-url*è la posizione del CloudFormation modello per la tua regione:

Region	Ubicazione del CloudFormation modello
Stati Uniti orientali (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Ubicazione del CloudFormation modello
Stati Uniti orientali (Virginia settentrionale)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Stati Uniti occidentali (California settentrionale)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Stati Uniti occidentali (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Canada (Centrale)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Europa (Irlanda)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Europa (Londra)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Europa (Parigi)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Ubicazione del CloudFormation modello
Regione Europa (Francoforte)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione di Israele (Tel Aviv)	`http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Asia Pacifico (Hong Kong)	`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`
Regione Asia Pacifico (Seoul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Asia Pacifico (Singapore)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Ubicazione del CloudFormation modello
Regione Asia Pacifico (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Asia Pacifico (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Asia Pacifico (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Regione Sud America (San Paolo)	`aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`

Questo comando crea uno CloudFormation stack denominato**`CodeDeployDemoStack`**, utilizzando il CloudFormation modello nel bucket Amazon S3 specificato. L'istanza Amazon EC2 è basata sul tipo di istanza t1.micro, ma è possibile utilizzare qualsiasi tipo. Presenta un tag applicato con il valore **`CodeDeployDemo`**, ma è possibile applicare un tag con qualsiasi valore. Ha la coppia di chiavi di istanza specificata applicata.

2. Chiama il describe-stacks comando per verificare che lo CloudFormation stack denominato **`CodeDeployDemoStack`** sia stato creato correttamente:

```
aws cloudformation describe-stacks --stack-name CodeDeployDemoStack --query
 "Stacks[0].StackStatus" --output text
```

Non continuare finché il valore `CREATE_COMPLETE` non viene restituito.

Per verificare che l' CodeDeploy agente sia in esecuzione sull'istanza Amazon EC2Gestione delle operazioni degli CodeDeploy agenti, consulta e procedi con. Crea un'applicazione con CodeDeploy

Configura un'istanza Amazon EC2 con cui lavorare CodeDeploy

Queste istruzioni mostrano come configurare un'istanza Amazon EC2 che esegue Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) o Windows Server da utilizzare nelle distribuzioni. CodeDeploy

> **Note**
>
> Se non disponi di un'istanza Amazon EC2, puoi utilizzare il CloudFormation modello per avviarne una che esegua Amazon Linux o Windows Server. Non forniamo un modello per Ubuntu Server o RHEL.

Fase 1: verifica che un profilo di istanza IAM sia collegato alla tua istanza Amazon EC2

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Nel riquadro di navigazione, in Istanze scegli Istanze.
3. Cerca e scegli la tua istanza Amazon EC2 nell'elenco.
4. Nel riquadro dei dettagli, nella scheda Descrizione, annota il valore nel campo del ruolo IAM, quindi passa alla sezione successiva.

   Se il campo è vuoto, puoi allegare un profilo di istanza IAM all'istanza. Per informazioni, consulta Associare un ruolo IAM a un'istanza.

Passaggio 2: verifica che il profilo dell'istanza IAM collegato disponga delle autorizzazioni di accesso corrette

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.

3. Naviga e scegli il nome del ruolo IAM che hai annotato nel passaggio 4 della sezione precedente.

   > **Note**
   >
   > Se desideri utilizzare il ruolo di servizio generato dal CloudFormation modello anziché uno creato seguendo le istruzioni riportate in Fase 2: Creare un ruolo di servizio per CodeDeploy, tieni presente quanto segue:
   > In alcune versioni del nostro CloudFormation modello, il nome visualizzato del profilo dell'istanza IAM generato e collegato alle istanze Amazon EC2 non è lo stesso del nome visualizzato nella console IAM. Ad esempio, il profilo dell'istanza IAM potrebbe avere un nome visualizzato di `CodeDeploySampleStack-expnyi6-InstanceRoleInstanceProfile-IK8J8A9123EX`, mentre il profilo dell'istanza IAM nella console IAM potrebbe avere un nome visualizzato di. `CodeDeploySampleStack-expnyi6-InstanceRole-C5P33V1L64EX`
   > Per aiutarti a identificare il profilo dell'istanza nella console IAM, vedrai che il prefisso di `CodeDeploySampleStack-expnyi6-InstanceRole` è lo stesso per entrambi.
   > Per informazioni sul motivo per cui questi nomi visualizzati potrebbero essere diversi, consulta Profili di istanza.

4. Scegli la scheda Relazioni di attendibilità. Se non è presente alcuna voce in Trusted Entities che riporta The identity provider (s) ec2.amazonaws.com, non puoi utilizzare questa istanza Amazon EC2. Interrompi e crea un'istanza Amazon EC2 utilizzando le informazioni contenute in. Utilizzo delle istanze per CodeDeploy

   Se c'è una voce che dice The identity provider (s) ec2.amazonaws.com e stai archiviando le tue applicazioni solo in GitHub repository, passa direttamente a. Fase 3: etichettare l'istanza Amazon EC2

   Se è presente una voce che dice The identity provider (s) ec2.amazonaws.com e stai archiviando le tue applicazioni in bucket Amazon S3, scegli la scheda Autorizzazioni.

5. Se è presente una policy nell'area Policy di autorizzazioni, scegliere il nome della policy e quindi selezionare Modifica policy.

6. Scegli la scheda JSON. Se stai archiviando le tue applicazioni in bucket Amazon S3, assicurati `"s3:Get*"` di essere nell'elenco delle azioni `"s3:List*"` specificate.

   Potrebbe essere simile a quanto segue:

```
{"Statement":[{"Resource":"*","Action":[
  ... Some actions may already be listed here ...
  "s3:Get*","s3:List*"
  ... Some more actions may already be listed here ...
  ],"Effect":"Allow"}]}
```

O potrebbe essere simile a quanto segue:

JSON

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

Se `"s3:Get*"` e non `"s3:List*"` sono nell'elenco di azioni specificate, scegliere Edit (Modifica) per aggiungerli, quindi scegliere Save (Salva). (Se né `"s3:Get*"` né `"s3:List*"` sono l'ultima operazione nell'elenco, assicurarsi di aggiungere una virgola dopo l'operazione, in modo da convalidare il documento della policy.)

> **Note**
>
> Ti consigliamo di limitare questa politica solo ai bucket Amazon S3 a cui devono accedere le tue istanze Amazon EC2. Assicurati di consentire l'accesso ai bucket Amazon S3 che contengono l'agente. CodeDeploy In caso contrario, potrebbe verificarsi un errore quando l' CodeDeploy agente viene installato o aggiornato sulle istanze.
> Per concedere al profilo dell'istanza IAM l'accesso solo ad alcuni bucket del kit di CodeDeploy risorse in Amazon S3, utilizza la seguente policy, ma rimuovi le righe relative ai bucket a cui desideri impedire l'accesso:

```
{
  "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/*"
      ]
    }
  ]
```

```
}
```

Fase 3: etichettare l'istanza Amazon EC2

Per istruzioni su come etichettare l'istanza Amazon EC2 in modo che CodeDeploy possa trovarla durante una distribuzione, consulta Lavorare con i tag nella console, quindi torna a questa pagina.

> **Note**
>
> Puoi etichettare l'istanza Amazon EC2 con qualsiasi chiave e valore desideri. Assicurarsi solo di specificare la chiave e il valore quando si distribuisce.

Passaggio 4: installa il AWS CodeDeploy agente sull'istanza Amazon EC2

Per istruzioni su come installare l' CodeDeploy agente sull'istanza Amazon EC2 e verificare che sia in esecuzioneGestione delle operazioni degli CodeDeploy agenti, consulta e procedi a. Crea un'applicazione con CodeDeploy

Utilizzo di istanze locali per CodeDeploy

Un'istanza locale è qualsiasi dispositivo fisico diverso da un'istanza Amazon EC2 in grado di eseguire CodeDeploy l'agente e connettersi agli endpoint del AWS servizio pubblico.

La distribuzione di una revisione CodeDeploy dell'applicazione su un'istanza locale prevede due passaggi principali:

- Passaggio 1: configurare ogni istanza locale, registrarla e quindi CodeDeploy etichettarla.
- Passaggio 2: distribuisci le revisioni dell'applicazione nell'istanza locale.

> **Note**
>
> Per provare a creare e a distribuire una revisione di un'applicazione di esempio a un'istanza locale configurata e registrata correttamente, consulta Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux). Per informazioni sulle istanze locali e su come funzionano, consulta. CodeDeploy Working with On-Premises Instances

Se non desideri più utilizzare un'istanza locale nelle distribuzioni, puoi rimuovere i tag delle istanze locali dai gruppi di distribuzione. In alternativa, puoi rimuovere i tag di istanza locale dall'istanza. O ancora, puoi annullare in modo esplicito la registrazione di un'istanza locale in modo che non possa più essere utilizzata in nessuna distribuzione. Per ulteriori informazioni, consulta Gestione delle operazioni delle istanze locali in CodeDeploy.

Le istruzioni in questa sezione mostrano come configurare un'istanza locale, quindi registrarla e contrassegnarla in CodeDeploy modo che possa essere utilizzata nelle distribuzioni. Questa sezione descrive anche come utilizzare per CodeDeploy ottenere informazioni sulle istanze locali e annullare la registrazione di un'istanza locale dopo che non si prevede più di distribuirla.

Argomenti

- Prerequisiti per la configurazione di un'istanza locale
- Registra un'istanza locale con CodeDeploy
- Gestione delle operazioni delle istanze locali in CodeDeploy

Prerequisiti per la configurazione di un'istanza locale

È necessario soddisfare i seguenti prerequisiti prima di poter annullare la registrazione di un'istanza locale.

> **Important**
>
> Se utilizzi il comando register-on-premises-instance e aggiorni periodicamente le credenziali temporanee generate con (), esistono altri prerequisiti. Servizio di token di sicurezza AWS AWS STS Per informazioni, consulta Prerequisiti per la registrazione ARN della sessione IAM.

Requisiti per il dispositivo

Il dispositivo che desideri preparare, registrare ed etichettare come istanza locale deve eseguire un sistema operativo supportato. CodeDeploy Per un elenco, consulta Sistemi operativi supportati dall'agente CodeDeploy.

Se il tuo sistema operativo non è supportato, l' CodeDeploy agente è disponibile come open source per adattarlo alle tue esigenze. Per ulteriori informazioni, consulta il repository degli CodeDeploy agenti in GitHub.

Comunicazione in uscita

L'istanza locale deve essere in grado di connettersi agli endpoint del AWS servizio pubblico con cui comunicare. CodeDeploy

L' CodeDeploy agente comunica in uscita utilizzando HTTPS sulla porta 443.

Controllo amministrativo

L'account locale o di rete utilizzato nell'istanza locale per configurare l'istanza locale deve poter essere eseguito come o `root` (per Ubuntu Server) `sudo` o come amministratore (per Windows Server).

Autorizzazioni IAM

All'identità IAM utilizzata per registrare l'istanza locale devono essere concesse le autorizzazioni per completare la registrazione (e per annullare la registrazione dell'istanza locale, se necessario).

Oltre alla policy descritta inPassaggio 3: Limita le autorizzazioni CodeDeploy dell'utente, assicurati che all'identità IAM chiamante sia allegata la seguente policy aggiuntiva.

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": "*"
    }
```

```
  ]
}
```

Per informazioni su come collegare le policy IAM, consulta Gestione di policy IAM.

Registra un'istanza locale con CodeDeploy

Per eseguire la registrazione di un'istanza locale, devi utilizzare un'identità IAM per autenticare le richieste. Puoi scegliere tra le seguenti opzioni per l'identità IAM e il metodo di registrazione che utilizzi:

- Utilizzo di un ARN ruolo IAM per autenticare le richieste.
  - Utilizza il comando register-on-premises-instance e aggiorna periodicamente le credenziali temporanee generate con () per configurare manualmente la maggior parte delle opzioni di registrazione. Servizio di token di sicurezza AWS AWS STS Questa opzione offre il massimo livello di sicurezza, poiché l'autenticazione avviene utilizzando un token temporaneo che scade e deve essere aggiornato periodicamente. Questa opzione è consigliata per implementazioni di produzione di qualsiasi dimensione. Per informazioni, consulta Utilizza il comando register-on-premises-instance (ARN della sessione IAM) per registrare un'istanza locale.
- (Non consigliato) Utilizza un ARN utente IAM per autenticare le richieste.
  - Utilizza il comando register per il processo di registrazione più automatizzato. Questa opzione dovrebbe essere utilizzata solo per le implementazioni non di produzione in cui la sicurezza è meno importante. Questa opzione è meno sicura perché utilizza credenziali statiche (permanenti) per l'autenticazione. Questa opzione è ideale per registrare una singola istanza locale. Per informazioni, consulta Usa il comando register (utente IAM ARN) per registrare un'istanza locale.
  - Utilizzate il comando register-on-premises-instance per configurare manualmente la maggior parte delle opzioni di registrazione. È un'opzione indicata per la registrazione di un piccolo numero di istanze locali. Per informazioni, consulta Usa il comando register-on-premises-instance (utente IAM ARN) per registrare un'istanza locale.

Argomenti

- Utilizza il comando register-on-premises-instance (ARN della sessione IAM) per registrare un'istanza locale
- Usa il comando register (utente IAM ARN) per registrare un'istanza locale
- Usa il comando register-on-premises-instance (utente IAM ARN) per registrare un'istanza locale

Utilizza il comando register-on-premises-instance (ARN della sessione IAM) per registrare un'istanza locale

Per il massimo controllo sull'autenticazione e la registrazione delle istanze locali, puoi utilizzare il comando register-on-premises-instance e aggiornare periodicamente le credenziali temporanee generate con (). Servizio di token di sicurezza AWS AWS STS Un ruolo IAM statico per l'istanza assume il ruolo di queste credenziali aggiornate per eseguire operazioni di distribuzione. AWS STS CodeDeploy

Questo metodo è molto utile quando è necessario registrare un numero elevato di istanze, Consente di automatizzare il processo di registrazione con. CodeDeploy È possibile utilizzare il proprio sistema di identità e autenticazione per autenticare le istanze locali e distribuire le credenziali di sessione IAM dal servizio alle istanze con cui utilizzarle. CodeDeploy

> **Note**
>
> In alternativa, puoi utilizzare un utente IAM condiviso distribuito su tutte le istanze locali per chiamare l' AWS STS AssumeRoleAPI e recuperare le credenziali di sessione per le istanze locali. Questo metodo è meno sicuro e non è consigliato per ambienti di produzione o mission-critical.

Utilizza le informazioni contenute nei seguenti argomenti per configurare un'istanza locale utilizzando credenziali di sicurezza temporanee generate con. AWS STS

Argomenti

- Prerequisiti per la registrazione ARN della sessione IAM
- Fase 1: Creare il ruolo IAM che le istanze locali assumeranno
- Passaggio 2: generare credenziali temporanee per una singola istanza utilizzando AWS STS
- Passaggio 3: aggiungere un file di configurazione all'istanza locale
- Fase 4: Preparare un'istanza locale per le distribuzioni CodeDeploy
- Passaggio 5: registrare l'istanza locale con CodeDeploy
- Passaggio 6: assegna un tag all'istanza locale
- Passaggio 7: distribuisci le revisioni dell'applicazione nell'istanza locale
- Fase 8: Tieni traccia delle distribuzioni sull'istanza locale

Prerequisiti per la registrazione ARN della sessione IAM

Oltre ai prerequisiti elencati in Prerequisiti per la configurazione di un'istanza locale, devono essere soddisfatti i seguenti requisiti aggiuntivi:

Autorizzazioni IAM

All'identità IAM utilizzata per registrare un'istanza locale devono essere concesse le autorizzazioni per eseguire operazioni. CodeDeploy Accertarti che la policy gestita AWSCodeDeployFullAccess sia collegata all'identità IAM. Per informazioni, consulta le policy AWS gestite nella Guida per l'utente IAM.

Sistema per aggiornare credenziali temporanee

Se utilizzi un ARN della sessione IAM per registrare istanze locali, è necessario disporre di un sistema per aggiornare periodicamente le credenziali temporanee. Le credenziali temporanee scadono dopo un'ora o prima se viene specificato un periodo di tempo più breve quando le credenziali vengono generate. Sono disponibili due metodi per aggiornare le credenziali:

- Metodo 1: utilizzare il sistema di identità e autenticazione operante nella rete aziendale con uno script CRON che effettua periodicamente il polling del sistema di identità e autenticazione e che copia le credenziali di sessione più recenti nell'istanza. Ciò consente di integrare la struttura di autenticazione e identità AWS senza dover apportare modifiche all' CodeDeploy agente o al servizio per supportare i tipi di autenticazione utilizzati nell'organizzazione.
- Metodo 2: esegui periodicamente un processo CRON sull'istanza per richiamare l' AWS STS AssumeRoleazione e scrivere le credenziali della sessione in un file a cui l' CodeDeploy agente può accedere. Questo metodo richiede ancora l'utilizzo di un utente IAM e la copia delle credenziali nell'istanza locale, ma è possibile riutilizzare lo stesso utente e le stesse credenziali IAM all'interno del parco istanze locali.

Per informazioni sulla creazione e l'utilizzo AWS STS delle credenziali, consulta Servizio di token di sicurezza AWS API Reference e Utilizzo delle credenziali di sicurezza temporanee per richiedere l'accesso alle risorse. AWS

Fase 1: Creare il ruolo IAM che le istanze locali assumeranno

Puoi utilizzare la console IAM AWS CLI o la console IAM per creare un ruolo IAM che verrà utilizzato dalle istanze locali per l'autenticazione e l'interazione. CodeDeploy

È sufficiente creare un singolo ruolo IAM. Ognuna delle istanze locali può assumere questo ruolo per recuperare le credenziali di sicurezza temporanee che forniscono le autorizzazioni concesse a questo ruolo.

Il ruolo che crei richiederà le seguenti autorizzazioni per accedere ai file necessari per installare l'agente: CodeDeploy

JSON

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

Ti consigliamo di limitare questa politica solo ai bucket Amazon S3 a cui deve accedere l'istanza locale. Se limiti questa politica, assicurati di consentire l'accesso ai bucket Amazon S3 che contengono l'agente. CodeDeploy In caso contrario, potrebbe verificarsi un errore ogni volta che l' CodeDeploy agente viene installato o aggiornato sull'istanza locale. Per informazioni sul controllo dell'accesso ai bucket Amazon S3, consulta Gestione delle autorizzazioni di accesso alle risorse Amazon S3.

Per creare il ruolo IAM

1. Chiamare il comando create-role utilizzando l'opzione `--role-name` per specificare un nome per il ruolo IAM (ad esempio, `CodeDeployInstanceRole`) e l'opzione `--assume-role-policy-document` per fornire le autorizzazioni.

   Quando si crea il ruolo IAM per questa istanza, è possibile denominarlo `CodeDeployInstanceRole` e includere le autorizzazioni richieste in un file denominato `CodeDeployRolePolicy.json`:

```
aws iam create-role --role-name CodeDeployInstanceRole --assume-role-policy-
document file://CodeDeployRolePolicy.json
```

2. Nell'output della chiamata al comando create-role, prendere nota del valore del campo ARN. Ad esempio:

   ```
   arn:aws:iam::123456789012:role/CodeDeployInstanceRole
   ```

   Avrai bisogno del ruolo ARN quando utilizzi l' AWS STS AssumeRoleAPI per generare credenziali a breve termine per ogni istanza.

   Per ulteriori informazioni sulla creazione di ruoli IAM, consulta Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'utente IAM.

   Per informazioni sull'assegnazione delle autorizzazioni a un ruolo esistente, consulta put-role-policy in Command Reference.AWS CLI

Passaggio 2: generare credenziali temporanee per una singola istanza utilizzando AWS STS

Prima di generare le credenziali temporanee che verranno utilizzate per registrare un'istanza locale, occorre creare o scegliere l'identità IAM (utente o ruolo) per la quale le credenziali temporanee verranno generate. L'autorizzazione `sts:AssumeRole` deve essere inclusa nelle impostazioni della policy per questa identità IAM.

Per informazioni sulla concessione `sts:AssumeRole` delle autorizzazioni a un'identità IAM, consulta Creazione di un ruolo per delegare le autorizzazioni a un servizio e. AWS AssumeRole

Sono disponibili due modi per generare le credenziali temporanee:

- Usa il comando assume-role con. AWS CLI Ad esempio:

  ```
  aws sts assume-role --role-arn arn:aws:iam::12345ACCOUNT:role/role-arn --role-
  session-name session-name
  ```

  Dove:

  - *12345ACCOUNT*è il numero di conto a 12 cifre dell'organizzazione.
  - *role-arn*è l'ARN del ruolo da assumere, in cui è stato generato. Fase 1: Creare il ruolo IAM che le istanze locali assumeranno

- *session-name* è il nome che si desidera assegnare alla sessione di ruolo che si sta creando ora.

> **Note**
>
> Se utilizzi uno script CRON che esegue periodicamente il polling del sistema di identità e autenticazione e copia le credenziali di sessione più recenti sull'istanza (metodo 1 per l'aggiornamento delle credenziali temporanee descritto in Prerequisiti per la registrazione ARN della sessione IAM), puoi invece utilizzare qualsiasi SDK supportato per la chiamata. AWS AssumeRole

- Utilizza uno strumento fornito da. AWS

  Lo strumento aws-codedeploy-session-helper genera AWS STS credenziali e le scrive in un file che inserisci nell'istanza. Questo strumento è il più appropriato per il metodo 2 per aggiornare le credenziali temporanee descritte in Prerequisiti per la registrazione ARN della sessione IAM. In questo metodo, lo strumento aws-codedeploy-session-helper viene posizionato su ogni istanza ed esegue il comando utilizzando le autorizzazioni di un utente IAM. Ogni istanza utilizza le stesse credenziali dell'utente IAM insieme a questo strumento.

  Per ulteriori informazioni, consulta il repository GitHub aws-codedeploy-session-helper.

> **Note**
>
> Dopo aver creato le credenziali della sessione IAM, posizionarle in qualsiasi posizione dell'istanza locale. Nel passaggio successivo, configurerai l'agente per accedere alle credenziali in questa posizione. CodeDeploy

Prima di continuare, assicurati che il sistema utilizzato per aggiornare periodicamente le credenziali temporanee sia disponibile. Se le credenziali temporanee non vengono aggiornate, le distribuzioni all'istanza locale non vanno a buon fine. Per ulteriori informazioni, consulta "Sistema per aggiornare le credenziali temporanee" in Prerequisiti per la registrazione ARN della sessione IAM.

Passaggio 3: aggiungere un file di configurazione all'istanza locale

Aggiungi un file di configurazione all'istanza locale, utilizzando le autorizzazioni dell'utente root o amministratore. Questo file di configurazione viene utilizzato per dichiarare le credenziali IAM e la AWS regione di destinazione per cui utilizzare. CodeDeploy Il file deve essere aggiunto in una

posizione specifica dell'istanza locale. Il file deve includere l'ARN della sessione temporanea IAM, l'ID della chiave segreta e la chiave di accesso segreta e la regione di destinazione AWS .

Per aggiungere un file di configurazione

1. Crea un file denominato `codedeploy.onpremises.yml` (per un'istanza locale di Ubuntu Server o RHEL) o `conf.onpremises.yml` (per un'istanza locale di Windows Server) nella seguente posizione sull'istanza locale:

   - Per Ubuntu Server: `/etc/codedeploy-agent/conf`
   - Per Windows Server: `C:\ProgramData\Amazon\CodeDeploy`

2. Utilizzate un editor di testo per aggiungere le seguenti informazioni al `codedeploy.onpremises.yml` file (Linux) o al `conf.onpremises.yml` file (Windows) appena creato:

   ```
   ---
   iam_session_arn: iam-session-arn
   aws_credentials_file: credentials-file
   region: supported-region
   ```

   Dove:

   - *`iam-session-arn`*è l'ARN della sessione IAM in cui hai annotato. Passaggio 2: generare credenziali temporanee per una singola istanza utilizzando AWS STS
   - *`credentials-file`*è la posizione del file delle credenziali per l'ARN della sessione temporanea, come indicato in. Passaggio 2: generare credenziali temporanee per una singola istanza utilizzando AWS STS
   - *`supported-region`*è una delle regioni CodeDeploy supportate, come elencato in Regione ed endpoint in. Riferimenti generali di AWS

Fase 4: Preparare un'istanza locale per le distribuzioni CodeDeploy

Installa e configura il AWS CLI

Installa e configura AWS CLI l'istanza locale. (AWS CLI Verrà utilizzato per scaricare e installare l' CodeDeploy agente sull'istanza locale.)

1. Per installare l' AWS CLI istanza locale, segui le istruzioni riportate nella Guida per l'utente AWS CLI nella Guida per l'AWS Command Line Interface utente.

   > **Note**
   >
   > CodeDeploy i comandi per lavorare con le istanze locali sono diventati disponibili nella versione 1.7.19 di. AWS CLI Se hai una versione di quella AWS CLI già installata, puoi verificarne la versione chiamando. aws --version

2. Per configurare l' AWS CLI istanza locale, segui le istruzioni in Configurazione della istanza locale nella Guida AWS CLI per l'AWS Command Line Interface utente.

   > **Important**
   >
   > Durante la configurazione AWS CLI (ad esempio, chiamando il aws configure comando), assicurati di specificare l'ID della chiave segreta e la chiave di accesso segreta di un utente IAM che disponga almeno delle autorizzazioni descritte in. Prerequisiti per la registrazione ARN della sessione IAM

Impostazione della variabile di ambiente AWS_REGION (solo server Ubuntu e RHEL)

Se non utilizzi Ubuntu Server o RHEL sulla tua istanza locale, salta questo passaggio e vai direttamente a «Installa l'agente». CodeDeploy

Installa l' CodeDeploy agente su un'istanza locale di Ubuntu Server o RHEL e abilita l'istanza ad aggiornare l' CodeDeploy agente ogni volta che diventa disponibile una nuova versione. Puoi farlo impostando la variabile di `AWS_REGION` ambiente sull'istanza sull'identificatore di una delle regioni supportate da. CodeDeploy Ti consigliamo di impostare il valore sulla regione in cui si trovano le CodeDeploy applicazioni, i gruppi di distribuzione e le revisioni delle applicazioni (ad esempio,`us-west-2`). Per un elenco di regioni, consulta Regione ed endpoint in. Riferimenti generali di AWS

Per impostare la variabile di ambiente, esegui la chiamata seguente dal terminale:

```
export AWS_REGION=supported-region
```

*`supported-region`*Dov'è l'identificatore della regione (ad esempio,`us-west-2`).

Installa l'agente CodeDeploy

- Per un'istanza locale di Ubuntu Server, segui le istruzioni riportate inInstalla l' CodeDeploy agente per Ubuntu Server, quindi torna a questa pagina.
- Per un'istanza RHEL locale, segui le istruzioni riportate inInstalla l' CodeDeploy agente per Amazon Linux o RHEL, quindi torna a questa pagina.
- Per un'istanza locale di Windows Server, segui le istruzioni riportate e torna a questa pagina. Installa l' CodeDeploy agente per Windows Server

Passaggio 5: registrare l'istanza locale con CodeDeploy

Le istruzioni in questa fase presumono che la registrazione dell'istanza locale venga eseguita dall'istanza locale stessa. È possibile registrare un'istanza locale da un dispositivo o un'istanza separato su cui è AWS CLI installata e configurata.

Utilizza il AWS CLI per registrare l'istanza locale CodeDeploy in modo che possa essere utilizzata nelle distribuzioni.

Prima di poter utilizzare AWS CLI, è necessario l'ARN delle credenziali di sessione temporanea in cui sono state create. Passaggio 3: aggiungere un file di configurazione all'istanza locale Ad esempio, per un'istanza identificata come `AssetTag12010298EX`:

```
arn:sts:iam::123456789012:assumed-role/CodeDeployInstanceRole/AssetTag12010298EX
```

Chiamate il comando register-on-premises-instance, specificando:

- Un nome che identifica in modo univoco l'istanza locale (con l'opzione `--instance-name`).

> **Important**
>
> Per identificare più facilmente l'istanza locale, soprattutto per scopi di debug, ti consigliamo di specificare un nome mappato ad alcune caratteristiche univoche dell'istanza locale (ad esempio, session-name delle credenziali STS e il numero di serie o un identificatore di asset interno, se applicabile). Se specificate un indirizzo MAC come nome, tenete presente che gli indirizzi MAC contengono caratteri CodeDeploy non consentiti, come i due punti (:). Per un elenco dei caratteri consentiti, consulta CodeDeploy quote.

- L'ARN della sessione IAM impostata per autenticare più istanze locali in Fase 1: Creare il ruolo IAM che le istanze locali assumeranno.

Ad esempio:

```
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
```

Dove:

- *name-of-instance*è il nome che usi per identificare l'istanza locale, ad esempio. `AssetTag12010298EX`
- *account-id*è l'ID dell'account a 12 cifre dell'organizzazione, ad esempio. `111222333444`
- *role-to-assume*è il nome del ruolo IAM che hai creato per l'istanza, ad esempio. `CodeDeployInstanceRole`
- *session-name*è il nome del ruolo di sessione in cui hai specificatoPassaggio 2: generare credenziali temporanee per una singola istanza utilizzando AWS STS.

Passaggio 6: assegna un tag all'istanza locale

Puoi utilizzare la AWS CLI o la CodeDeploy console per etichettare l'istanza locale. (CodeDeployutilizza i tag delle istanze locali per identificare gli obiettivi di distribuzione durante una distribuzione.)

Per applicare tag all'istanza locale (CLI)

- Chiamate il comando add-tags-to-on-premises-instances, specificando:
  - Il nome che identifica in modo univoco l'istanza locale (con l'opzione `--instance-names`).
  - Il nome della chiave tag e del valore tag dell'istanza locale che si desidera utilizzare (con l'opzione `--tags`). È necessario specificare sia un nome che un valore. CodeDeploy non consente tag di istanza locali con solo valori.

    Ad esempio:

    ```
    aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
     --tags Key=Name,Value=CodeDeployDemo-OnPrem
    ```

Per applicare tag all'istanza locale (console)

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy e scegli On-premises istanze.
3. Nell'elenco delle istanze locali, scegliere il nome dell'istanza locale a cui desideri applicare tag.
4. Nell'elenco di tag, selezionare o immettere la chiave tag e il valore tag desiderati. Dopo aver immesso la chiave tag e il valore tag, viene visualizzata un'altra riga. È possibile ripetere questa operazione fino a 10 tag. Per rimuovere un tag, scegli Remove (Rimuovi).
5. Dopo aver aggiunto i tag, scegliere Update Tags (Aggiorna tag).

Passaggio 7: distribuisci le revisioni dell'applicazione nell'istanza locale

È ora possibile distribuire le revisioni dell'applicazione all'istanza locale registrata e contrassegnata.

Distribuisci le revisioni delle applicazioni sulle istanze locali in un modo simile alla distribuzione delle revisioni delle applicazioni sulle istanze Amazon EC2. Per istruzioni, consulta Crea una distribuzione con CodeDeploy. Queste istruzioni includono un link ai prerequisiti, inclusa la creazione di un'applicazione, la creazione di un gruppo di distribuzione e la preparazione di una revisione dell'applicazione. Se occorre una semplice revisione dell'applicazione di esempio da distribuire, puoi creare quella descritta in Passaggio 2: creare una revisione di esempio dell'applicazione nel Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux).

> **Important**
>
> Se riutilizzi un ruolo di CodeDeploy servizio come parte della creazione di un gruppo di distribuzione destinato alle istanze locali, devi includere `Tag:get*` nella parte della politica del ruolo di servizio. `Action` Per ulteriori informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.

Fase 8: Tieni traccia delle distribuzioni sull'istanza locale

Dopo aver distribuito una revisione dell'applicazione a istanze locali registrate e contrassegnate, puoi monitorare l'avanzamento della distribuzione.

Tieni traccia delle distribuzioni su istanze locali in modo simile al monitoraggio delle distribuzioni su istanze Amazon EC2. Per istruzioni, consulta Visualizza i dettagli CodeDeploy della distribuzione.

Usa il comando register (utente IAM ARN) per registrare un'istanza locale

> **Important**
>
> La registrazione di un'istanza utilizzando un utente IAM non è consigliata perché utilizza credenziali statiche (permanenti) per l'autenticazione. Per una maggiore sicurezza, consigliamo di registrare un'istanza utilizzando credenziali temporanee per l'autenticazione. Per ulteriori informazioni, consulta Utilizza il comando register-on-premises-instance (ARN della sessione IAM) per registrare un'istanza locale.

> **Important**
>
> Assicurati di disporre di un piano per la rotazione delle chiavi di accesso dell'utente IAM (credenziali permanenti). Per ulteriori informazioni, consulta Rotazione delle chiavi di accesso.

Questa sezione descrive come configurare, registrare e applicare tag a un'istanza locale con il minimo sforzo utilizzando CodeDeploy . Il comando register è particolarmente utile quando si utilizza una singola istanza locale o un parco di istanze locali di piccole dimensioni. È possibile utilizzare il register comando solo quando si utilizza un ARN utente IAM per autenticare un'istanza. Non è possibile utilizzare il register comando con un ARN di sessione IAM per l'autenticazione.

Quando usi il register comando, puoi lasciare CodeDeploy fare quanto segue:

- Crea un utente IAM AWS Identity and Access Management per l'istanza locale, se non ne specifichi uno con il comando.
- Salva le credenziali dell'utente IAM in un file di configurazione dell'istanza locale.
- Registra l'istanza locale con. CodeDeploy
- Aggiungere tag all'istanza locale, se li specifichi come parte del comando.

> **Note**
>
> Il comando register-on-premises-instance è un'alternativa al comando register. https://docs.aws.amazon.com/cli/latest/reference/deploy/register.html Il comando register-on-premises-instance è indicato se desideri configurare, registrare e aggiungere tag a un'istanza locale con CodeDeploy per la maggior parte in modo autonomo. Il register-on-premises-instance comando offre anche la possibilità di utilizzare un ARN di sessione IAM per registrare le istanze anziché un ARN utente IAM. Questo approccio offre un vantaggio notevole se disponi di parchi di istanze locali di grandi dimensioni. In particolare, puoi utilizzare un singolo ARN di sessione IAM per autenticare più istanze invece di dover creare un utente IAM per ogni istanza locale uno per una. Per ulteriori informazioni, consulta Usa il comando register-on-premises-instance (utente IAM ARN) per registrare un'istanza locale e Utilizza il comando register-on-premises-instance (ARN della sessione IAM) per registrare un'istanza locale.

Argomenti

- Passaggio 1: installare e configurare il AWS CLI nell'istanza locale
- Passaggio 2: richiama il comando register
- Passaggio 3: richiama il comando di installazione
- Passaggio 4: distribuzione delle revisioni dell'applicazione nell'istanza locale
- Passaggio 5: traccia delle distribuzioni sull'istanza locale

Passaggio 1: installare e configurare il AWS CLI nell'istanza locale

1. Installa AWS CLI l'istanza locale. Segui le istruzioni riportate nella Guida per l'utente AWS CLI nella Guida per l'AWS Command Line Interface utente.

   > **Note**
   >
   > CodeDeploy i comandi per lavorare con le istanze locali sono disponibili nella AWS CLI versione 1.7.19 e successive. Se hai AWS CLI già installato, chiama aws --version per verificarne la versione.

2. Configura AWS CLI l'istanza locale. Segui le istruzioni riportate nella Guida alla configurazione della Guida per l'AWS Command Line Interface utente AWS CLI in.

Important

Mentre configuri AWS CLI (ad esempio, chiamando il aws configure comando), assicurati di specificare l'ID della chiave segreta e la chiave di accesso segreta di un utente IAM che dispone almeno delle seguenti autorizzazioni di AWS accesso oltre alle autorizzazioni specificate in. Prerequisiti per la configurazione di un'istanza locale In questo modo è possibile scaricare e installare l' CodeDeploy agente sull'istanza locale. Le autorizzazioni di accesso potrebbero essere simili a queste:

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:*",
        "iam:CreateAccessKey",
        "iam:CreateUser",
        "iam:DeleteAccessKey",
        "iam:DeleteUser",
        "iam:DeleteUserPolicy",
        "iam:ListAccessKeys",
        "iam:ListUserPolicies",
        "iam:PutUserPolicy",
        "iam:GetUser",
        "tag:getTagKeys",
        "tag:getTagValues",
        "tag:GetResources"
      ],
      "Resource" : "*"
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource" : [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
```

```
        "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    }
  ]
}
```

> **Note**
>
> Se visualizzi errori di accesso negato quando provi ad accedere a uno dei bucket Amazon S3 mostrati in precedenza, prova a omettere la `/*` parte dell'ARN delle risorse del bucket, ad esempio. `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/*"
      ]
    }
  ]
}
```

Passaggio 2: richiama il comando register

Per questa fase, si presuppone che la registrazione dell'istanza locale venga eseguita dall'istanza locale stessa. Puoi anche registrare un'istanza locale da un dispositivo o un'istanza separata su cui è AWS CLI installato e configurato come descritto nel passaggio precedente.

Usa il comando AWS CLI per chiamare il registro, specificando:

- Un nome che identifica in modo univoco l'istanza locale di CodeDeploy (con l'opzione). `--instance-name`

  > **Important**
  >
  > Per identificare più facilmente l'istanza locale in un secondo momento, soprattutto a scopo di debug, è consigliabile utilizzare un nome mappato ad alcune caratteristiche univoche dell'istanza locale (ad esempio, il numero di serie o un identificatore di asset interno, se applicabili). Se specificate un indirizzo MAC per un nome, tenete presente che gli indirizzi MAC contengono caratteri CodeDeploy non consentiti, come i due punti (). `:` Per un elenco dei caratteri consentiti, consulta CodeDeploy quote.

- Facoltativamente, l'ARN di un utente IAM esistente che desideri associare a questa istanza locale (con l'`--iam-user-arn`opzione). Per ottenere l'ARN di un utente IAM, chiama il comando get-user o scegli il nome utente IAM nella sezione Users della console IAM, quindi trova il valore User ARN nella sezione Summary. Se questa opzione non è specificata, CodeDeploy creerà un utente IAM per tuo conto nel tuo AWS account e lo assocerà all'istanza locale.

> **Important**
>
> Se specifichi l'opzione `--iam-user-arn`, devi anche creare manualmente il file di configurazione dell'istanza locale, come descritto in Passaggio 4: aggiungere un file di configurazione all'istanza locale.
> Puoi associare un solo utente IAM a una sola istanza locale. Il tentativo di associare un singolo utente IAM a più istanze locali può causare errori, distribuzioni non riuscite a tali istanze locali o distribuzioni a quelle istanze locali bloccate in uno stato di sospeso perpetuo.

- Facoltativamente, un set di tag di istanza locali (con l'`--tags`opzione) che CodeDeploy verranno utilizzati per identificare il set di istanze Amazon EC2 su cui distribuire. Specifica ciascun tag con `Key=`*`tag-key`*`,Value=`*`tag-value`* (ad esempio, `Key=Name,Value=Beta Key=Name,Value=WestRegion`). Se questa opzione non è specificata, non verrà registrato alcun tag. Per registrare i tag in un secondo momento, chiama il comando add-tags-to-on-premises-instances. https://docs.aws.amazon.com/cli/latest/reference/deploy/add-tags-to-on-premises-instances.html
- Facoltativamente, la AWS regione in cui verrà registrata l'istanza locale (con l'opzione). CodeDeploy `--region` Questa deve essere una delle regioni supportate elencate in Regione e gli endpoint in Riferimenti generali di AWS (ad esempio,). `us-west-2` Se questa opzione non è specificata, verrà utilizzata la AWS regione predefinita associata all'utente IAM chiamante.

Ad esempio:

```
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
```

Il comando register funziona nel modo seguente:

1. Se non viene specificato alcun utente IAM esistente, crea un utente IAM, gli assegna le autorizzazioni richieste e genera una chiave segreta e una chiave di accesso segreta corrispondenti. L'istanza locale utilizzerà questo utente IAM e le relative autorizzazioni e credenziali per autenticarsi e interagire con esso. CodeDeploy
2. Registra l'istanza locale con. CodeDeploy

3. Se specificato, associa CodeDeploy i tag specificati all'`--tags`opzione al nome dell'istanza locale registrata.
4. Se è stato creato un utente IAM, crea anche il file di configurazione richiesto nella stessa directory da cui è stato chiamato il register comando.

Se questo comando rileva errori, viene visualizzato un messaggio di errore che descrive in che modo è possibile completare manualmente i passaggi rimanenti. In caso contrario, viene visualizzato un messaggio che descrive come chiamare il comando install, come illustrato nella fase successiva.

Passaggio 3: richiama il comando di installazione

Dall'istanza locale, usa il comando AWS CLI to call install, specificando:

- Il percorso del file di configurazione (con l'opzione `--config-file`).
- Facoltativamente, se sostituire il file di configurazione già presente nell'istanza locale (con l'opzione `--override-config`). Se non specificato, il file di configurazione esistente non verrà sostituito.
- Facoltativamente, la AWS regione in cui verrà registrata l'istanza locale CodeDeploy (con l'opzione). `--region` Questa deve essere una delle regioni supportate elencate in Regione e gli endpoint in Riferimenti generali di AWS (ad esempio,). `us-west-2` Se questa opzione non è specificata, verrà utilizzata la AWS regione predefinita associata all'utente IAM chiamante.
- Facoltativamente, una posizione personalizzata da cui installare l' CodeDeploy agente (con l'`--agent-installer`opzione). Questa opzione è utile per installare una versione personalizzata dell' CodeDeploy agente che non CodeDeploy supporta ufficialmente (ad esempio una versione personalizzata basata sul repository dell'CodeDeployagente in GitHub). Il valore deve essere il percorso di un bucket Amazon S3 che contiene:
  - Uno script di installazione CodeDeploy dell'agente (per Linux o sistemi Unix-based operativi, simile al file di installazione nel repository degli CodeDeploy agenti in). GitHub
  - Un file del pacchetto di installazione CodeDeploy dell'agente (.msi) (per Windows-based i sistemi operativi).

  Se questa opzione non è specificata, CodeDeploy farà del suo meglio per installare dalla propria postazione una versione ufficialmente supportata dell' CodeDeploy agente compatibile con il sistema operativo sull'istanza locale.

Ad esempio:

```
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
```

Il comando install funziona nel modo seguente:

1. Verifica se l'istanza locale è un'istanza Amazon EC2. In caso affermativo, viene visualizzato un messaggio di errore.
2. Copia il file di configurazione delle istanze locali dalla posizione specificata sull'istanza alla posizione in cui l' CodeDeploy agente prevede di trovarlo, a condizione che il file non si trovi già in quella posizione.

   Per Ubuntu Server e Red Hat Enterprise Linux (RHEL)), questo è/. `/etc/codedeploy-agent/conf codedeploy.onpremises.yml`

   Per Windows Server, questo è`C:\ProgramData\Amazon \CodeDeploy\conf.onpremises.yml`.

   Se l'opzione `--override-config` è stata specificata, crea o sovrascrive il file.
3. Installa l' CodeDeploy agente sull'istanza locale e quindi lo avvia.

Passaggio 4: distribuzione delle revisioni dell'applicazione nell'istanza locale

È ora possibile distribuire le revisioni dell'applicazione all'istanza locale registrata e contrassegnata.

Le revisioni delle applicazioni vengono distribuite su istanze locali in modo simile alla distribuzione delle revisioni delle applicazioni su istanze Amazon EC2. Per istruzioni, consulta l'argomento Crea una distribuzione con CodeDeploy. Queste istruzioni rimandano ai prerequisiti, inclusa la creazione di un'applicazione, la creazione di un gruppo di distribuzione e la preparazione di una revisione di un'applicazione. Se occorre una semplice revisione dell'applicazione di esempio da distribuire, puoi creare quella descritta in Passaggio 2: creare una revisione di esempio dell'applicazione nel Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux).

> **Important**
>
> Se riutilizzi un ruolo di CodeDeploy servizio esistente come parte della creazione di un gruppo di distribuzione destinato alle istanze locali, devi includerlo nella parte della

dichiarazione politica del ruolo di servizio. `Tag:get*` `Action` Per ulteriori informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.

Passaggio 5: traccia delle distribuzioni sull'istanza locale

Dopo aver distribuito una revisione dell'applicazione a istanze locali registrate e contrassegnate, puoi monitorare l'avanzamento della distribuzione.

Tieni traccia delle distribuzioni sulle istanze locali in modo simile al monitoraggio delle distribuzioni sulle istanze Amazon EC2. Per istruzioni, consulta l'argomento Visualizza i dettagli CodeDeploy della distribuzione.

Per ulteriori opzioni, consulta Gestione delle operazioni delle istanze locali in CodeDeploy.

Usa il comando register-on-premises-instance (utente IAM ARN) per registrare un'istanza locale

Important

La registrazione di un'istanza utilizzando un utente IAM non è consigliata perché utilizza credenziali statiche (permanenti) per l'autenticazione. Per una maggiore sicurezza, consigliamo di registrare un'istanza utilizzando credenziali temporanee per l'autenticazione. Per ulteriori informazioni, consulta Utilizza il comando register-on-premises-instance (ARN della sessione IAM) per registrare un'istanza locale.

Important

Assicurati di disporre di un piano per la rotazione delle chiavi di accesso dell'utente IAM (credenziali permanenti). Per ulteriori informazioni, consulta Rotazione delle chiavi di accesso.

Segui le istruzioni per configurare un'istanza locale, registrarla e applicarvi tag con CodeDeploy in modo prevalentemente autonomo utilizzando le credenziali utente IAM per l'autenticazione.

Argomenti

- Passaggio 1: creare un utente IAM per l'istanza locale

- Fase 2: Assegna le autorizzazioni all'utente IAM
- Fase 3: Ottieni le credenziali utente IAM
- Passaggio 4: aggiungere un file di configurazione all'istanza locale
- Fase 5: Installare e configurare AWS CLI
- Passaggio 6: imposta la variabile di ambiente AWS_REGION (solo Ubuntu Server e RHEL)
- Fase 7: Installare l'agente CodeDeploy
- Passaggio 8: registrare l'istanza locale con CodeDeploy
- Passaggio 9: contrassegnare l'istanza locale
- Passaggio 10: distribuzione delle revisioni delle applicazioni nell'istanza locale
- Passaggio 11: traccia delle distribuzioni sull'istanza locale

Passaggio 1: creare un utente IAM per l'istanza locale

Crea un utente IAM che l'istanza locale utilizzerà per autenticarsi e interagire con. CodeDeploy

> **Important**
>
> È necessario creare un utente IAM separato per ogni istanza locale partecipante. Se provi a riutilizzare un singolo utente IAM per più istanze locali, potresti non riuscire a registrare o etichettare correttamente tali istanze locali con. CodeDeploy Le distribuzioni in tali istanze locali potrebbero restare bloccate in uno stato di sospensione perpetuo o avere esito negativo.

Ti consigliamo di assegnare all'utente IAM un nome che ne identifichi lo scopo, ad esempio. CodeDeployUser-OnPrem

Puoi utilizzare la AWS CLI o la console IAM per creare un utente IAM. Per informazioni, consulta Creazione di un utente IAM nel tuo AWS account.

> **Important**
>
> Sia che utilizzi la console IAM AWS CLI o la console IAM per creare un nuovo utente IAM, prendi nota dell'ARN utente fornito all'utente. Queste informazioni ti serviranno in seguito in Passaggio 4: aggiungere un file di configurazione all'istanza locale e Passaggio 8: registrare l'istanza locale con CodeDeploy.

Fase 2: Assegna le autorizzazioni all'utente IAM

Se la tua istanza locale distribuirà le revisioni delle applicazioni dai bucket Amazon S3, devi assegnare all'utente IAM le autorizzazioni per interagire con tali bucket. Puoi utilizzare la AWS CLI o la console IAM per assegnare le autorizzazioni.

> **Note**
>
> Se intendi distribuire le revisioni delle applicazioni solo dai GitHub repository, salta questo passaggio e vai direttamente a. Fase 3: Ottieni le credenziali utente IAM (Avrai comunque bisogno di informazioni sull'utente IAM in cui hai creato. Passaggio 1: creare un utente IAM per l'istanza locale Verrà utilizzato nei passaggi successivi.)

Per assegnare le autorizzazioni (CLI)

1. Crea un file con i seguenti contenuti della policy sull'istanza o sul dispositivo Amazon EC2 che stai utilizzando per chiamare il AWS CLI. Assegnare al file un nome tipo **`CodeDeploy-OnPrem-Permissions.json`** e salvarlo.

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

> **Note**
>
> Ti consigliamo di limitare questa policy solo ai bucket Amazon S3 a cui la tua istanza locale deve accedere. Se limiti questa policy, assicurati di dare accesso anche ai bucket Amazon S3 che contengono l'agente. AWS CodeDeploy Altrimenti, potrebbe verificarsi

un errore ogni volta che l' CodeDeploy agente viene installato o aggiornato sull'istanza locale associata.
Ad esempio:

```
{
  "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. Chiama il comando put-user-policy, specificando il nome dell'utente IAM (con l'`--user-name`opzione), un nome per la policy (con l'`--policy-name`opzione) e il percorso del documento di policy appena creato (con l'opzione). `--policy-document` Ad esempio, presupponendo che il file **`CodeDeploy-OnPrem-Permissions.json`** si trovi nella stessa directory (cartella) da cui si sta chiamando il comando:

> **Important**
>
> Assicurarsi di includere `file://` prima del nome del file. Questo è obbligatorio in questo comando.

```
aws iam put-user-policy --user-name CodeDeployUser-OnPrem --policy-name CodeDeploy-OnPrem-Permissions --policy-document file://CodeDeploy-OnPrem-Permissions.json
```

Per assegnare le autorizzazioni (console)

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policies (Policy) e Create Policy (Crea policy). (Se viene visualizzato il pulsante Get Started (Inizia), sceglierlo, quindi scegliere Create Policy (Crea policy)).
3. Accanto a Create Your Own Policy (Crea la tua policy) scegli Select (Seleziona).
4. Nella casella Policy Name (Nome policy) digitare un nome per questa policy (ad esempio **`CodeDeploy-OnPrem-Permissions`**).
5. Nella casella Policy Document, digita o incolla la seguente espressione di autorizzazione, che consente di AWS CodeDeploy distribuire le revisioni delle applicazioni da qualsiasi bucket Amazon S3 specificato nella policy all'istanza locale per conto dell'utente IAM:

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

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

6. Scegli Crea policy.
7. Nel pannello di navigazione, scegliere Users (Utenti).
8. Nell'elenco degli utenti, cerca e scegli il nome dell'utente IAM in cui hai creato. Passaggio 1: creare un utente IAM per l'istanza locale
9. Nella scheda Permissions (Autorizzazioni), in Managed Policies (Policy gestite), selezionare Attach Policy (Collega policy).
10. Selezionare la policy denominata **`CodeDeploy-OnPrem-Permissions`**, quindi scegliere Attach Policy (Collega policy).

Fase 3: Ottieni le credenziali utente IAM

Ottieni l'ID della chiave segreta e la chiave di accesso segreta per l'utente IAM. che serviranno per Passaggio 4: aggiungere un file di configurazione all'istanza locale. Puoi utilizzare la AWS CLI o la console IAM per ottenere l'ID della chiave segreta e la chiave di accesso segreta.

> **Note**
>
> Se già si dispone dell'ID chiave segreta e dell'ID chiave di accesso segreta, saltare questo passaggio e passare direttamente a Passaggio 4: aggiungere un file di configurazione all'istanza locale.
> Gli utenti hanno bisogno di un accesso programmatico se vogliono interagire con l' AWS Console di gestione AWS esterno. Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede. AWS
> Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Consigliato) Utilizza le credenziali della console come credenziali temporanee per firmare richieste programmatiche agli AWS SDK o alle AWS CLI API. AWS	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per saperne di più AWS CLI, consulta Login for AWS local development nella Guida per l'utente. AWS Command Line Interface • Per AWS gli SDK, consulta Login for AWS local development nella Guida di riferimento per AWS SDK and Tools.

Quale utente necessita dell'accesso programmatico?	Per	Come
Identità della forza lavoro (Utenti gestiti nel centro identità IAM)	Usa le credenziali temporanee per firmare le richieste programmatiche agli AWS SDK o alle AWS CLI API. AWS	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per saperne di più AWS CLI, consulta Configurazione dell'uso Centro identità AWS IAM nella Guida AWS CLI per l'utente. AWS Command Line Interface • Per AWS SDK, strumenti e AWS API, consulta l'autenticazione IAM Identity Center nella Guida di riferimento per AWS SDK e strumenti.
IAM	Usa le credenziali temporanee per firmare le richieste programmatiche agli SDK o alle API AWS CLI. AWS AWS	Seguendo le istruzioni in Uso delle credenziali temporanee con le AWS risorse nella IAM User Guide.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Non consigliato) Usa le credenziali a lungo termine per firmare richieste programmatiche agli AWS CLI AWS SDK o alle API. AWS	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per saperne di più AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente. AWS Command Line Interface • Per AWS SDK e strumenti, consulta Autenticare utilizzando credenziali a lungo termine nella Guida di riferimento per SDK e strumenti. AWS • Per le AWS API, consulta Gestire le chiavi di accesso per gli utenti IAM nella IAM User Guide.

Per ottenere le credenziali (CLI)

1. Chiama il comando list-access-keys, specificando il nome dell'utente IAM (con l'`--user-name`opzione) e richiedendo solo gli ID delle chiavi di accesso (con le opzioni e). `--query` `--output` Ad esempio:

```
aws iam list-access-keys --user-name CodeDeployUser-OnPrem --query
 "AccessKeyMetadata[*].AccessKeyId" --output text
```

2. Se nell'output non appare alcuna chiave o nell'output compaiono informazioni su una sola chiave, chiama il comando create-access-key, specificando il nome dell'utente IAM (con l'opzione): `--user-name`

```
aws iam create-access-key --user-name CodeDeployUser-OnPrem
```

Nell'output della chiamata al comando create-access-key prendere nota del valore dei campi `AccessKeyId` e `SecretAccessKey`. Queste informazioni serviranno in Passaggio 4: aggiungere un file di configurazione all'istanza locale.

> **Important**
>
> Sarà l'unica volta che si avrà accesso a questa chiave di accesso segreta. Se si dimentica o si perde l'accesso a questa chiave di accesso segreta, sarà necessario generarne una nuova seguendo le fasi descritte in Fase 3: Ottieni le credenziali utente IAM.

3. Se sono già elencate due chiavi di accesso, è necessario eliminarne una chiamando il comando delete-access-key, specificando il nome dell'utente IAM (con l'`--user-name`opzione) e l'ID della chiave di accesso da eliminare (con l'`--access-key-id`opzione). Quindi chiamare il comando create-access-key, come descritto precedentemente in questa fase. Ecco un esempio di utilizzo del comando delete-access-key:

```
aws iam delete-access-key --user-name CodeDeployUser-OnPrem --access-key-id access-key-ID
```

> **Important**
>
> Se chiami il delete-access-key comando per eliminare una di queste chiavi di accesso e un'istanza locale sta già utilizzando questa chiave di accesso come descritto inPassaggio 4: aggiungere un file di configurazione all'istanza locale, dovrai seguire Passaggio 4: aggiungere un file di configurazione all'istanza locale nuovamente le istruzioni per specificare un ID della chiave di accesso e una chiave di accesso segreta diversi associati a questo utente IAM. In caso contrario, le distribuzioni in questa istanza locale potrebbero restare bloccate in uno stato di sospensione perpetuo o avere esito negativo.

Per ottenere le credenziali (console)

1. a. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
   b. Se l'elenco di utenti non è visualizzato, nel riquadro di navigazione scegliere Users (Utenti).
   c. Nell'elenco degli utenti, cerca e scegli il nome dell'utente IAM in Passaggio 1: creare un utente IAM per l'istanza locale cui hai creato.
2. Nella scheda Security credentials (Credenziali di sicurezza), se nell'elenco non sono presenti chiavi, o è presente una sola chiave, scegliere Create access key (Crea chiave di accesso).

   Se nell'elenco sono presenti due chiavi di accesso, è necessario eliminarne una. Selezionare Delete (Elimina) accanto a una delle chiavi di accesso, quindi scegliere Create access key (Crea chiave di accesso).

   > **Important**
   >
   > Se scegli Elimina accanto a una di queste chiavi di accesso e un'istanza locale sta già utilizzando questa chiave di accesso come descritto inPassaggio 4: aggiungere un file di configurazione all'istanza locale, dovrai seguire Passaggio 4: aggiungere un file di configurazione all'istanza locale nuovamente le istruzioni per specificare un ID della chiave di accesso e una chiave di accesso segreta diversi associati a questo utente IAM. In caso contrario, le distribuzioni in questa istanza locale potrebbero restare bloccate in uno stato di sospensione perpetuo o avere esito negativo.
3. Selezionare Show (Mostra) e prendere nota dell'ID chiave di accesso e della chiave di accesso segreta. Queste informazioni serviranno per la fase successiva. In alternativa, selezionare Download .csv file (Scarica file .csv) per salvare una copia dell'ID chiave di accesso e la chiave di accesso segreta.

   > **Important**
   >
   > Se non si prende nota o si scaricano le credenziali, questa sarà l'unica volta che si avrà accesso a questa chiave di accesso segreta. Se si dimentica o si perde l'accesso a questa chiave di accesso segreta, sarà necessario generarne una nuova seguendo le fasi descritte in Fase 3: Ottieni le credenziali utente IAM.
4. Scegli Chiudi per tornare alla *IAM User Name* pagina Utenti >.

Passaggio 4: aggiungere un file di configurazione all'istanza locale

Aggiungi un file di configurazione all'istanza locale, utilizzando le autorizzazioni dell'utente root o amministratore. Questo file di configurazione verrà utilizzato per dichiarare le credenziali utente IAM e la AWS regione di destinazione da utilizzare. CodeDeploy Il file deve essere aggiunto in una posizione specifica dell'istanza locale. Il file deve includere l'ARN dell'utente IAM, l'ID della chiave segreta, la chiave di accesso segreta e la regione di destinazione. AWS Il file deve seguire un formato specifico.

1. Crea un file denominato `codedeploy.onpremises.yml` (per un'istanza locale di Ubuntu Server o RHEL) o `conf.onpremises.yml` (per un'istanza locale di Windows Server) nella seguente posizione sull'istanza locale:

   - Per Ubuntu Server: `/etc/codedeploy-agent/conf`
   - Per Windows Server: `C:\ProgramData\Amazon\CodeDeploy`

2. Utilizzare un editor di testo per aggiungere le seguenti informazioni al nuovo file `codedeploy.onpremises.yml` o `conf.onpremises.yml` creato:

   ```
   ---
   aws_access_key_id: secret-key-id
   aws_secret_access_key: secret-access-key
   iam_user_arn: iam-user-arn
   region: supported-region
   ```

   Dove:

   - *`secret-key-id`*è l'ID della chiave segreta dell'utente IAM corrispondente che hai annotato in Passaggio 1: creare un utente IAM per l'istanza locale oFase 3: Ottieni le credenziali utente IAM.
   - *`secret-access-key`*è la chiave di accesso segreta dell'utente IAM corrispondente che hai annotato in Passaggio 1: creare un utente IAM per l'istanza locale oFase 3: Ottieni le credenziali utente IAM.
   - *`iam-user-arn`*è l'ARN dell'utente IAM che hai annotato in Passaggio 1: creare un utente IAM per l'istanza locale precedenza.
   - *`supported-region`*è l'identificatore di una regione supportata da CodeDeploy dove si trovano le CodeDeploy applicazioni, i gruppi di distribuzione e le revisioni delle applicazioni (ad esempio,). `us-west-2` Per un elenco delle regioni, consulta Regione ed endpoint in. Riferimenti generali di AWS

> **Important**
>
> Se hai scelto Elimina accanto a una delle chiavi di accesso in Fase 3: Ottieni le credenziali utente IAM e la tua istanza locale sta già utilizzando l'ID della chiave di accesso e la chiave di accesso segreta associati, dovrai seguire le istruzioni per specificare un ID della chiave di Passaggio 4: aggiungere un file di configurazione all'istanza locale accesso e una chiave di accesso segreta diversi associati a questo utente IAM. In caso contrario, le distribuzioni nella propria istanza locale potrebbero restare bloccate in uno stato di sospensione perpetuo o avere esito negativo.

Fase 5: Installare e configurare AWS CLI

Installa e configura AWS CLI l'istanza locale. (AWS CLI Verrà utilizzato Fase 7: Installare l'agente CodeDeploy per scaricare e installare l' CodeDeploy agente sull'istanza locale.)

1. Per installare l' AWS CLI istanza locale, segui le istruzioni riportate nella Guida all'installazione con la AWS CLI Guida per l'AWS Command Line Interface utente.

   > **Note**
   >
   > CodeDeploy i comandi per lavorare con le istanze locali sono diventati disponibili nella versione 1.7.19 di. AWS CLI Se disponi di una versione AWS CLI già installata, puoi verificarne la versione chiamando. aws --version

2. Per configurare l'istanza AWS CLI in locale, segui le istruzioni in Configurazione della AWS CLI nella Guida per l'AWS Command Line Interface utente.

   > **Important**
   >
   > Mentre configuri AWS CLI (ad esempio, chiamando il aws configure comando), assicurati di specificare l'ID della chiave segreta e la chiave di accesso segreta di un utente IAM che dispone almeno delle seguenti autorizzazioni di AWS accesso oltre alle autorizzazioni di accesso specificate nel. Prerequisiti per la configurazione di un'istanza locale Ciò consente di scaricare e installare l' CodeDeploy agente sull'istanza locale:

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" : [
        "*"
      ]
    }
  ]
}
```

Queste autorizzazioni di accesso possono essere assegnate all'utente IAM creato Passaggio 1: creare un utente IAM per l'istanza locale o a un altro utente IAM. Per assegnare queste autorizzazioni a un utente IAM, segui le istruzioni inPassaggio 1: creare un utente IAM per l'istanza locale, utilizzando queste autorizzazioni di accesso anziché quelle in quel passaggio.

Passaggio 6: imposta la variabile di ambiente AWS_REGION (solo Ubuntu Server e RHEL)

Se non stai utilizzando Ubuntu Server o RHEL sulla tua istanza locale, salta questo passaggio e vai direttamente a. Fase 7: Installare l'agente CodeDeploy

Installa l' CodeDeploy agente su un'istanza locale di Ubuntu Server o RHEL e abilita l'istanza ad aggiornare l' CodeDeploy agente ogni volta che diventa disponibile una nuova versione. Puoi farlo impostando la variabile di `AWS_REGION` ambiente sull'istanza sull'identificatore di una delle regioni supportate da. CodeDeploy Si consiglia di impostare il valore sulla regione in cui si trovano le CodeDeploy applicazioni, i gruppi di distribuzione e le revisioni delle applicazioni (ad esempio,). `us-west-2` Per un elenco delle regioni, consulta Regione ed endpoint in. Riferimenti generali di AWS

Per impostare la variabile di ambiente, esegui la chiamata seguente dal terminale:

```
export AWS_REGION=supported-region
```

*`supported-region`*Dov'è l'identificatore della regione (ad esempio,`us-west-2`).

Fase 7: Installare l'agente CodeDeploy

Installa l' CodeDeploy agente sull'istanza locale:

- Per un'istanza locale di Ubuntu Server, segui le istruzioni contenute in Installa l' CodeDeploy agente per Ubuntu Server, quindi torna a questa pagina.
- Per un'istanza RHEL locale, segui le istruzioni contenute in Installa l' CodeDeploy agente per Amazon Linux o RHEL, quindi torna a questa pagina.
- Per un'istanza locale di Windows Server, segui le istruzioni contenute in Installa l' CodeDeploy agente per Windows Server, quindi torna a questa pagina.

Passaggio 8: registrare l'istanza locale con CodeDeploy

Le istruzioni in questa fase presumono che la registrazione dell'istanza locale venga eseguita dall'istanza locale stessa. È possibile registrare un'istanza locale da un dispositivo o istanza separato su cui è AWS CLI installata e configurata, come descritto in. Fase 5: Installare e configurare AWS CLI

Usa il AWS CLI per registrare l'istanza locale CodeDeploy in modo che possa essere utilizzata nelle distribuzioni.

1. Prima di poter utilizzare il AWS CLI, avrai bisogno dell'ARN utente dell'utente IAM in cui hai creato. Passaggio 1: creare un utente IAM per l'istanza locale Se non disponi già dell'ARN utente, chiama il comando get-user, specificando il nome dell'utente IAM (con l'`--user-name`opzione) e interrogando solo l'ARN dell'utente (con le opzioni and): `--query` `--output`

   ```
   aws iam get-user --user-name CodeDeployUser-OnPrem --query "User.Arn" --output text
   ```

2. Chiama il comando register-on-premises-instance, specificando: https://docs.aws.amazon.com/cli/latest/reference/deploy/register-on-premises-instance.html

   - Un nome che identifica in modo univoco l'istanza locale (con l'opzione `--instance-name`).

     > **Important**
     >
     > Per identificare più facilmente l'istanza locale, soprattutto a scopo di debug, consigliamo di specificare un nome mappato ad alcune caratteristiche univoche dell'istanza locale (ad esempio, il numero di serie o un identificatore di asset interno, se applicabili). Se specificate un indirizzo MAC come nome, tenete presente che gli indirizzi MAC contengono caratteri CodeDeploy non consentiti, come i due punti (). `:` Per un elenco dei caratteri consentiti, consulta CodeDeploy quote.

- L'ARN utente dell'utente IAM in cui hai creato Passaggio 1: creare un utente IAM per l'istanza locale (con l'`--iam-user-arn`opzione).

  Ad esempio:

```
aws deploy register-on-premises-instance --instance-name AssetTag12010298EX --
iam-user-arn arn:aws:iam::444455556666:user/CodeDeployUser-OnPrem
```

Passaggio 9: contrassegnare l'istanza locale

Puoi utilizzare la console AWS CLI o la CodeDeploy console per etichettare l'istanza locale. (CodeDeployutilizza i tag delle istanze locali per identificare gli obiettivi di distribuzione durante una distribuzione.)

Per applicare tag all'istanza locale (CLI)

- Chiama il comando https://docs.aws.amazon.com/cli/latest/reference/deploy/add-tags-to-on-premises-instances.html add-tags-to-on-premises-instances, specificando:

  - Il nome che identifica in modo univoco l'istanza locale (con l'opzione `--instance-names`).
  - Il nome della chiave tag e del valore tag dell'istanza locale che si desidera utilizzare (con l'opzione `--tags`). È necessario specificare sia un nome che un valore. CodeDeploy non consente tag di istanza locali contenenti solo valori.

  Ad esempio:

```
aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
 --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Per applicare tag all'istanza locale (console)

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Dal CodeDeploy menu, scegli On-premises istanze.
3. Nell'elenco delle istanze locali scegliere la freccia accanto all'istanza locale a cui si desidera applicare tag.
4. Nell'elenco di tag, selezionare o immettere la chiave tag e il valore tag desiderati. Dopo aver immesso la chiave tag e il valore tag, viene visualizzata un'altra riga. È possibile ripetere questa operazione fino a 10 tag. Per rimuovere un tag, scegliere l'icona di eliminazione ( ).
5. Dopo aver aggiunto i tag, scegliere Update Tags (Aggiorna tag).

Passaggio 10: distribuzione delle revisioni delle applicazioni nell'istanza locale

È ora possibile distribuire le revisioni dell'applicazione all'istanza locale registrata e contrassegnata.

Le revisioni delle applicazioni vengono distribuite su istanze locali in modo simile alla distribuzione delle revisioni delle applicazioni su istanze Amazon EC2. Per istruzioni, consulta l'argomento Crea una distribuzione con CodeDeploy. Queste istruzioni includono un link ai prerequisiti, inclusa la creazione di un'applicazione, la creazione di un gruppo di distribuzione e la preparazione di una revisione dell'applicazione. Se occorre una semplice revisione dell'applicazione di esempio da distribuire, puoi creare quella descritta in Passaggio 2: creare una revisione di esempio dell'applicazione nel Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux).

> **Important**
>
> Se riutilizzi un ruolo di CodeDeploy servizio come parte della creazione di un gruppo di distribuzione destinato alle istanze locali, devi includerlo nella parte della dichiarazione politica del ruolo di servizio. `Tag:get* Action` Per ulteriori informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.

Passaggio 11: traccia delle distribuzioni sull'istanza locale

Dopo aver distribuito una revisione dell'applicazione a istanze locali registrate e contrassegnate, puoi monitorare l'avanzamento della distribuzione.

Tieni traccia delle distribuzioni sulle istanze locali in modo simile al monitoraggio delle distribuzioni sulle istanze Amazon EC2. Per istruzioni, consulta Visualizza i dettagli CodeDeploy della distribuzione.

Gestione delle operazioni delle istanze locali in CodeDeploy

Segui le istruzioni in questa sezione per gestire le operazioni sulle istanze locali dopo averle registrate CodeDeploy, ad esempio per ottenere maggiori informazioni, rimuovere i tag e disinstallare e annullare la registrazione delle istanze locali.

Argomenti

- Ottieni informazioni su una singola istanza locale
- Ottieni informazioni su più istanze locali
- Rimuovi manualmente i tag delle istanze locali da un'istanza locale
- Disinstalla automaticamente l' CodeDeploy agente e rimuovi il file di configurazione da un'istanza locale
- Annulla automaticamente la registrazione di un'istanza locale
- Annulla manualmente la registrazione di un'istanza locale

Ottieni informazioni su una singola istanza locale

È possibile ottenere informazioni su una singola istanza locale seguendo le istruzioni in Visualizza i dettagli CodeDeploy della distribuzione. Puoi utilizzare la console AWS CLI o la CodeDeploy console per ottenere ulteriori informazioni su una singola istanza locale.

Per ottenere informazioni su una singola istanza locale (CLI)

- Chiama il comando get-on-premises-instance, specificando il nome che identifica in modo univoco l'istanza locale (con l'opzione): `--instance-name`

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

Per ottenere informazioni su una singola istanza locale (console)

1. Accedi a e apri la console all'indirizzo. Console di gestione AWS CodeDeploy https://console.aws.amazon.com/codedeploy

> **Note**
>
> Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy e scegli On-premises istanze.
3. Nell'elenco delle istanze locali, scegliere il nome di un'istanza locale per visualizzarne i dettagli.

Ottieni informazioni su più istanze locali

È possibile ottenere informazioni sulle istanze locali seguendo le istruzioni in Visualizza i dettagli CodeDeploy della distribuzione. Puoi utilizzare la console AWS CLI o la CodeDeploy console per ottenere maggiori informazioni sulle istanze locali.

Per ottenere informazioni su più istanze locali (CLI)

1. Per un elenco di nomi di istanze locali, chiama il comando list-on-premises-instances, specificando:

   - Se si desidera ottenere informazioni su tutte le istanze locali registrate o con registrazione annullata (rispettivamente con l'opzione `--registration-status` e `Registered` o `Deregistered`). Se si omette questa informazione, vengono restituiti i nomi delle istanze locali sia registrate sia con registrazione annullata.
   - Se si desidera ottenere informazioni solo sulle istanze locali a cu sono applicati tag di istanze locali specifici (con l'opzione `--tag-filters`). Per ogni tag di istanza locale, specificare `Key`, `Value` e `Type` (che deve essere sempre `KEY_AND_VALUE`). Separare più tag di istanze locali inserendo uno spazio tra una terna formata da `Key`, `Value` e `Type` e quella successiva.

   Ad esempio:

   ```
   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. Per informazioni più dettagliate, chiamate il comando batch-get-on-premises-instances, con i nomi delle istanze locali (con l'opzione): `--instance-names`

   ```
   aws deploy batch-get-on-premises-instances --instance-names AssetTag12010298EX
    AssetTag09920444EX
   ```

Per ottenere informazioni su più istanze locali (console)

1. Accedi a e apri la console all'indirizzo. Console di gestione AWS CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy e scegli On-premises istanze.

   Vengono visualizzate le informazioni sulle istanze locali.

Rimuovi manualmente i tag delle istanze locali da un'istanza locale

Generalmente si rimuove un tag che non viene più utilizzato da un'istanza locale oppure si rimuove l'istanza locale dai gruppi di distribuzione che si basano su quel tag. Puoi utilizzare la console AWS CLI o la AWS CodeDeploy console per rimuovere i tag delle istanze locali dalle istanze locali.

Non è necessario eliminare i tag da un'istanza locale prima di annullare la registrazione.

La rimozione manuale dei tag da un'istanza locale non ne annulla la registrazione. Non disinstalla l' CodeDeploy agente dall'istanza. Non rimuove il file di configurazione dall'istanza. Non elimina l'utente IAM associato all'istanza.

Per annullare automaticamente la registrazione dell'istanza locale, consulta Annulla automaticamente la registrazione di un'istanza locale.

Per annullare manualmente la registrazione dell'istanza locale, consulta Annulla manualmente la registrazione di un'istanza locale.

Per disinstallare automaticamente l' CodeDeploy agente e rimuovere il file di configurazione dall'istanza locale, consultaDisinstalla automaticamente l' CodeDeploy agente e rimuovi il file di configurazione da un'istanza locale.

Per disinstallare manualmente solo l' CodeDeploy agente dall'istanza locale, vedi. Gestione delle operazioni degli CodeDeploy agenti

Per eliminare manualmente l'utente IAM associato, consulta Eliminazione di un utente IAM dal tuo AWS account.

Per rimuovere i tag da un'istanza locale (CLI)

- Chiama remove-tags-from-on-premises-instances, specificando:

  - I nomi che identificano in modo univoco l'istanza locale (con l'opzione `--instance-names`).
  - I nomi e i valori dei tag che si desidera rimuovere (con l'opzione `--tags`).

  Ad esempio:

```
aws deploy remove-tags-from-on-premises-instances --instance-names
 AssetTag12010298EX --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Per rimuovere i tag da un'istanza locale (console)

1. Accedi a e apri la console all' Console di gestione AWS indirizzo. CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy e scegli On-premises istanze.
3. Nell'elenco delle istanze locali, scegliere il nome dell'istanza locale da cui si desidera rimuovere i tag.
4. Nella sezione Tags (Tag) scegliere Remove (Rimuovi) accanto a ogni tag che si desidera rimuovere.
5. Dopo avere eliminato i tag, scegliere Update tags (Aggiorna tag).

Disinstalla automaticamente l' CodeDeploy agente e rimuovi il file di configurazione da un'istanza locale

In genere, si disinstalla l' CodeDeploy agente e si rimuove il file di configurazione da un'istanza locale dopo che non si prevede più di distribuirla.

> **Note**
>
> La disinstallazione automatica dell' CodeDeploy agente e la rimozione del file di configurazione da un'istanza locale non annullano la registrazione di un'istanza locale. Non elimina l'associazione dei tag associati all'istanza locale. Non elimina l'utente IAM associato all'istanza locale.
> Per annullare automaticamente la registrazione dell'istanza locale, consulta Annulla automaticamente la registrazione di un'istanza locale.
> Per annullare manualmente la registrazione dell'istanza locale, consulta Annulla manualmente la registrazione di un'istanza locale.
> Per annullare manualmente l'associazione di tag di istanze locali associati, consulta Rimuovi manualmente i tag delle istanze locali da un'istanza locale.
> Per disinstallare manualmente l' CodeDeploy agente dall'istanza locale, consulta. Gestione delle operazioni degli CodeDeploy agenti
> Per eliminare manualmente l'utente IAM associato, consulta Eliminazione di un utente IAM dal tuo AWS account.

Dall'istanza locale, usa il comando AWS CLI to call il comando uninstall.

Ad esempio:

```
aws deploy uninstall
```

Il comando uninstall funziona nel modo seguente:

1. Arresta l' CodeDeploy agente in esecuzione sull'istanza locale.
2. Disinstalla l' CodeDeploy agente dall'istanza locale.
3. Rimuove il file di configurazione dall'istanza locale. (Per Ubuntu Server e RHEL, questo è/. `/etc/codedeploy-agent/conf` `codedeploy.onpremises.yml` Per Windows Server, questo è`C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.)

Annulla automaticamente la registrazione di un'istanza locale

Generalmente si annulla la registrazione di un'istanza locale quando non si ha più intenzione di distribuire in quell'istanza. Quando si annulla la registrazione di un'istanza locale, anche se l'istanza locale potrebbe essere parte dei tag delle istanze locali di un gruppo di distribuzione, l'istanza locale

non viene inclusa in alcuna distribuzione. Puoi utilizzarlo per annullare la registrazione delle AWS CLI istanze locali.

> **Note**
>
> Non è possibile utilizzare la CodeDeploy console per annullare la registrazione di un'istanza locale. Inoltre, l'annullamento della registrazione di un'istanza locale rimuove l'associazione dei tag a tale istanza locale. Non disinstalla l' CodeDeploy agente dall'istanza locale. Non rimuove il file di configurazione dell'istanza locale dall'istanza.
> Per utilizzare la CodeDeploy console per eseguire alcune (ma non tutte) le attività di questa sezione, consulta la sezione dedicata alla CodeDeploy console diAnnulla manualmente la registrazione di un'istanza locale.
> Per annullare manualmente l'associazione di tag di istanze locali associati, consulta Rimuovi manualmente i tag delle istanze locali da un'istanza locale.
> Per disinstallare automaticamente l' CodeDeploy agente e rimuovere il file di configurazione dall'istanza locale, vedereDisinstalla automaticamente l' CodeDeploy agente e rimuovi il file di configurazione da un'istanza locale.
> Per disinstallare manualmente solo l' CodeDeploy agente dall'istanza locale, vedi. Gestione delle operazioni degli CodeDeploy agenti

Utilizzate il comando AWS CLI per chiamare il comando deregister, specificando:

- Il nome che identifica in modo univoco l'istanza locale a CodeDeploy (con l'opzione). `--instance-name`
- Facoltativamente, se eliminare l'utente IAM associato all'istanza locale. Il comportamento predefinito consiste nell'eliminare l'utente IAM. Se non si desidera eliminare l'utente IAM associato all'istanza locale, specificare l'opzione `--no-delete-iam-user` nel comando.
- Facoltativamente, la AWS regione in cui è stata registrata l'istanza locale CodeDeploy (con l'`--region`opzione). Deve essere una delle regioni supportate elencate in Regione e gli endpoint in Riferimenti generali di AWS(ad esempio,). `us-west-2` Se questa opzione non è specificata, verrà utilizzata la AWS regione predefinita associata all'utente IAM chiamante.

Un esempio di annullamento della registrazione di un'istanza ed eliminazione dell'utente:

```
aws deploy deregister --instance-name AssetTag12010298EX --region us-west-2
```

Un esempio di annullamento della registrazione di un'istanza senza eliminazione dell'utente:

```
aws deploy deregister --instance-name AssetTag12010298EX --no-delete-iam-user --region
 us-west-2
```

Il comando deregister funziona nel modo seguente:

1. Annulla la registrazione dell'istanza locale con. CodeDeploy
2. Se specificato, elimina l'utente IAM associato all'istanza locale.

Dopo aver annullato la registrazione di un'istanza locale:

- L'istanza non viene più visualizzata nella console sin da subito.
- È possibile creare immediatamente un'altra istanza con lo stesso nome.

Se questo comando rileva errori, viene visualizzato un messaggio di errore che descrive in che modo è possibile completare manualmente i passaggi rimanenti. In caso contrario, viene visualizzato un messaggio che descrive come chiamare il comando uninstall.

Annulla manualmente la registrazione di un'istanza locale

Generalmente si annulla la registrazione di un'istanza locale quando non si ha più intenzione di distribuire in quell'istanza. Lo usi per annullare manualmente la AWS CLI registrazione delle istanze locali.

L'annullamento manuale della registrazione di un'istanza locale non comporta la disinstallazione dell'agente. CodeDeploy Non rimuove il file di configurazione dall'istanza. Non elimina l'utente IAM associato all'istanza. Non rimuove i tag associati all'istanza.

Per disinstallare automaticamente l' CodeDeploy agente e rimuovere il file di configurazione dall'istanza locale, consultaDisinstalla automaticamente l' CodeDeploy agente e rimuovi il file di configurazione da un'istanza locale.

Per disinstallare manualmente solo l' CodeDeploy agente, consultaGestione delle operazioni degli CodeDeploy agenti.

Per eliminare manualmente l'utente IAM associato, consulta Eliminazione di un utente IAM dal tuo AWS account.

Per rimuovere manualmente solo i tag di istanze locali associati, consulta Rimuovi manualmente i tag delle istanze locali da un'istanza locale.

- Chiama il comando deregister-on-premises-instance, specificando il nome che identifica in modo univoco l'istanza locale (con l'opzione): `--instance-name`

  ```
  aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
  ```

  Dopo aver annullato la registrazione di un'istanza locale:

  - L'istanza non viene più visualizzata nella console sin da subito.
  - È possibile creare immediatamente un'altra istanza con lo stesso nome.

Visualizza i dettagli dell'istanza con CodeDeploy

Puoi utilizzare la CodeDeploy console AWS CLI, le o le CodeDeploy API per visualizzare i dettagli sulle istanze utilizzate in una distribuzione.

Per informazioni sull'utilizzo delle azioni CodeDeploy API per visualizzare le istanze, consulta GetDeploymentInstance, ListDeploymentInstancese. ListOnPremisesInstances

Argomenti

- Visualizza i dettagli dell'istanza (console)
- Visualizza i dettagli dell'istanza (CLI)

Visualizza i dettagli dell'istanza (console)

Per visualizzare i dettagli dell'istanza:

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy, quindi scegli Deployments.

> **Note**
>
> Se non viene visualizzata alcuna voce, assicurati che sia selezionata la regione corretta. Nella barra di navigazione, nel selettore di regione, scegli una delle regioni elencate in Regione ed endpoint in. Riferimenti generali di AWS CodeDeploy è supportato solo in queste regioni.

3. Per visualizzare i dettagli di distribuzione, scegliere l'ID distribuzione per l'istanza.
4. È possibile visualizzare tutte le istanze nella sezione Instance activity (Attività istanza) della pagina della distribuzione.
5. Per visualizzare informazioni sui singoli eventi del ciclo di vita della distribuzione per un'istanza, nella pagina dei dettagli della distribuzione, nella colonna Eventi, scegli Visualizza eventi.

> **Note**
>
> Se è visualizzato Failed (Non riuscito) per un evento del ciclo di vita, scegliere nella pagina dei dettagli dell'istanza View logs (Visualizza log), View in EC2 (Visualizza in EC2) o entrambe le opzioni. Suggerimenti per la risoluzione dei problemi sono disponibili in Risoluzione dei problemi relativi alle istanze.

6. Se desideri visualizzare ulteriori informazioni su un'istanza Amazon EC2, scegli l'ID dell'istanza nella colonna Instance ID.

Visualizza i dettagli dell'istanza (CLI)

Per utilizzare l'opzione AWS CLI per visualizzare i dettagli dell'istanza, chiamate il `get-deployment-instance` comando o il `list-deployment-instances` comando.

Per visualizzare i dettagli su una singola istanza, chiamate il comando get-deployment-instance, specificando:

- L'ID distribuzione univoco. Per ottenere l'ID di distribuzione, chiama il comando list-deployments.
- L'ID istanza univoco. Per ottenere l'ID dell'istanza, chiama il comando list-deployment-instances.

Per visualizzare un elenco di ID per le istanze utilizzate in una distribuzione, chiamate il comando list-deployment-instances, specificando:

- L'ID distribuzione univoco. Per ottenere l'ID di distribuzione, chiama il comando list-deployments.
- Eventualmente, stabilisci se includere solo specifici ID istanza in base allo stato di distribuzione. Se non specificato, vengono elencati tutti gli ID istanza corrispondenti, indipendentemente dallo stato di distribuzione.

CodeDeploy salute dell'istanza

CodeDeploy monitora lo stato di integrità delle istanze in un gruppo di distribuzione. Le distribuzioni non riescono se il numero di istanze integre è inferiore al numero minimo di istanze integre specificato per il gruppo di distribuzione durante la distribuzione. Ad esempio, se durante l'implementazione deve rimanere integro l'85% delle istanze e il gruppo di distribuzione contiene 10 istanze, è sufficiente che la distribuzione a una sola istanza non riesca per compromettere l'esito dell'intera distribuzione. Questo perché quando un'istanza viene messa offline in modo da poter installare l'ultima revisione dell'applicazione, il numero di istanze integre disponibili scende già al 90%. Pertanto, quando un'istanza non riesce e un'altra istanza è offline, solo l'80% delle istanze è integro e disponibile. CodeDeploy non riuscirà quindi a eseguire la distribuzione generale.

È importante ricordare che perché la distribuzione generale riesca, devono verificarsi le seguenti condizioni:

- CodeDeploy è in grado di eseguire la distribuzione su ogni istanza della distribuzione.
- La distribuzione ad almeno un'istanza ha esito positivo. Ciò significa che, anche se il valore minimo di istanze correttamente in esecuzione è 0, perché la distribuzione generale riesca, è necessario che la distribuzione di almeno una istanza abbia esito positivo (ovvero deve essere integra almeno un'istanza).

Argomenti

- Health status (Stato di integrità)
- Informazioni sul numero minimo di istanze integre
- Circa il numero minimo di istanze integre per zona di disponibilità

Health status (Stato di integrità)

CodeDeploy assegna due valori di stato di integrità a ciascuna istanza: integrità della revisione e integrità dell'istanza.

Integrità della revisione

L'integrità della revisione si basa sulla revisione dell'applicazione attualmente installata nell'istanza. Include i seguenti valori di stato:

- Corrente: la revisione installata nell'istanza corrisponde alla revisione per l'ultima distribuzione riuscita del gruppo di distribuzione.
- Precedente: la revisione installata nell'istanza corrisponde a una versione precedente dell'applicazione.
- Sconosciuta: la revisione dell'applicazione non è stata installata con successo nell'istanza.

Stato dell'istanza

L'integrità dell'istanza si basa sulla riuscita o meno delle distribuzioni a un'istanza. Include i seguenti valori:

- Integra: l'ultima distribuzione all'istanza ha avuto esito positivo.
- Non integra: il tentativo di distribuire una revisione all'istanza ha avuto esito negativo o non è stata ancora distribuita una revisione all'istanza.

CodeDeploy utilizza l'integrità della revisione e lo stato dell'istanza per pianificare la distribuzione nelle istanze del gruppo di distribuzione nel seguente ordine:

1. Istanza non integra.
2. Integrità revisione sconosciuta.
3. Revisione precedente.
4. Revisione corrente.

Se la distribuzione complessiva ha esito positivo, la revisione viene aggiornata e i valori dello stato di integrità del gruppo di distribuzione vengono aggiornati per riflettere l'ultima distribuzione.

- Un'istanza corrente interessata da una distribuzione riuscita mantiene lo stato corrente. In caso contrario, passa allo stato sconosciuta.
- Un'istanza precedente o sconosciuta interessata da una distribuzione riuscita passa allo stato corrente. In caso contrario, mantiene lo stato precedente o sconosciuta.
- Un'istanza integra interessata da una distribuzione riuscita mantiene lo stato integra. In caso contrario, passa allo stato non integra.

- Un'istanza non integra interessata da una distribuzione riuscita passa allo stato integra. In caso contrario, mantiene lo stato non integra.

Se la distribuzione complessiva ha esito negativo o viene interrotta:

- Per ogni istanza in cui si è CodeDeploy tentato di distribuire la revisione dell'applicazione lo stato dell'istanza è impostato su integro o non integro, a seconda che il tentativo di distribuzione per quell'istanza abbia avuto esito positivo o negativo.
- Ogni istanza in cui non CodeDeploy ha tentato di distribuire la revisione dell'applicazione mantiene il valore di integrità dell'istanza corrente.
- La revisione del gruppo di distribuzione resta la stessa.

Informazioni sul numero minimo di istanze integre

Il numero minimo di istanze integre richiesto viene definito come parte di una configurazione della distribuzione.

> **Important**
>
> Durante una blue/green distribuzione, la configurazione di distribuzione e il valore minimo degli host integri si applicano alle istanze nell'ambiente sostitutivo, non a quelle dell'ambiente originale. Tuttavia, quando viene annullata la registrazione delle istanze nell'ambiente originale dal sistema di bilanciamento del carico, la distribuzione complessiva viene contrassegnata come non riuscita anche se l'annullamento della registrazione non viene effettuato con successo per una sola istanza originale.

CodeDeploy fornisce tre configurazioni di distribuzione predefinite che utilizzano valori minimi di host integri di uso comune:

Nome della configurazione di distribuzione predefinita	Valori minimi per host integri predefiniti
CodeDeployDefault.OneAtATime	1
CodeDeployDefault.HalfAtATime	50%

Nome della configurazione di distribuzione predefinita	Valori minimi per host integri predefiniti
CodeDeployDefault.AllAtOnce	0

Ulteriori informazioni sulle configurazioni di distribuzione predefinite sono disponibili in Utilizzo delle configurazioni di distribuzione in CodeDeploy.

È possibile creare configurazioni di distribuzione personalizzate CodeDeploy per definire i propri valori minimi di integrità dell'host. Puoi definire questi valori come numeri interi o percentuali quando utilizzi le seguenti operazioni:

- Come `minimum-healthy-hosts` quando si utilizza il comando create-deployment-config in. AWS CLI
- Come nel tipo di dati `Value` nell'API. MinimumHealthyHosts CodeDeploy
- Come `MinimumHealthyHosts` quando usi AWS::CodeDeploy:: DeploymentConfig in un CloudFormation modello.

CodeDeploy consente di specificare un numero minimo di istanze integre per la distribuzione per due scopi principali:

- Per determinare se la distribuzione complessiva ha esito positivo o negativo. La distribuzione ha esito positivo se la revisione dell'applicazione è stata distribuita con successo ad almeno il numero minimo di istanze integre.
- Per determinare il numero di istanze che devono essere integre durante una distribuzione affinché quest'ultima possa proseguire.

Puoi specificare il numero minimo di istanze integre per il tuo gruppo di distribuzione come numero di istanze o come percentuale del numero totale di istanze. Se si specifica una percentuale, all'inizio della distribuzione, CodeDeploy converte la percentuale nel numero equivalente di istanze, arrotondando per eccesso le istanze frazionarie.

CodeDeploy tiene traccia dello stato di integrità delle istanze del gruppo di distribuzione durante il processo di distribuzione e utilizza il numero minimo di istanze integre specificato per la distribuzione per determinare se continuare la distribuzione. Il principio di base è che una distribuzione non deve far mai scendere il numero di istanze integre al di sotto del valore minimo specificato. L'unica

eccezione a questa regola può verificarsi quando un gruppo di distribuzione include inizialmente un numero minimo di istanze integre inferiore a quello minimo specificato. In quel caso, il processo di distribuzione non riduce ulteriormente il numero di istanze integre.

> **Note**
>
> CodeDeploy tenterà di eseguire la distribuzione su tutte le istanze di un gruppo di distribuzione, anche su quelle attualmente in uno stato Interrotto. Nel calcolo del numero minimo di istanze integre, un'istanza interrotta ha lo stesso impatto di un'istanza non riuscita. Per risolvere il problema di distribuzioni non riuscite a causa di un numero eccessivo di istanze interrotte, riavvia le istanze o modificane i tag per escluderle dal gruppo di distribuzione.

CodeDeploy avvia il processo di distribuzione tentando di distribuire la revisione dell'applicazione sulle istanze non integre del gruppo di distribuzione. Per ogni distribuzione riuscita, CodeDeploy modifica lo stato di integrità dell'istanza in integro e la aggiunge alle istanze integre del gruppo di distribuzione. CodeDeploy quindi confronta il numero attuale di istanze integre con il numero minimo specificato di istanze integre.

- Se il numero di istanze integre è inferiore o uguale al numero minimo specificato di istanze integre, CodeDeploy annulla la distribuzione per garantire che il numero di istanze integre non diminuisca con un numero maggiore di distribuzioni.
- Se il numero di istanze integre è superiore al numero minimo specificato di istanze integre di almeno una, CodeDeploy distribuisce la revisione dell'applicazione nel set originale di istanze integre.

Se una distribuzione su un'istanza integra fallisce, CodeDeploy modifica lo stato di integrità dell'istanza in non integro. Man mano che la distribuzione procede, CodeDeploy aggiorna il numero attuale di istanze integre e lo confronta con il numero minimo specificato di istanze integre. Se il numero di istanze integre scende al numero minimo specificato in qualsiasi momento del processo di distribuzione, interrompe la distribuzione. CodeDeploy Questa misura previene la possibilità che la prossima distribuzione abbia esito negativo facendo scendere il numero di istanze integre al di sotto di quello minimo specificato.

> **Note**
>
> Assicurati che il numero minimo di istanze integre che specifichi sia inferiore al numero totale di istanze nel gruppo di distribuzione. Se specifichi un valore percentuale, ricorda che verrà arrotondato. In caso contrario, quando la distribuzione viene avviata, il numero di istanze integre sarà già inferiore o uguale a quello minimo e CodeDeploy farà sì che la distribuzione complessiva abbia immediatamente esito negativo.

CodeDeploy utilizza inoltre il numero minimo specificato di istanze integre e il numero effettivo di istanze integre per determinare se e come distribuire la revisione dell'applicazione su più istanze. Per impostazione predefinita, CodeDeploy distribuisce la revisione dell'applicazione nel maggior numero possibile di istanze senza il rischio che il numero di istanze integre scenda al di sotto del numero minimo specificato di istanze integre.

Per determinare il numero di istanze su cui distribuire contemporaneamente, utilizza il seguente calcolo: CodeDeploy

```
[total-hosts] - [minimum-healthy-hosts] =
        [number-of-hosts-to-deploy-to-at-once]
```

Ad esempio:

- Se il gruppo di distribuzione ha 10 istanze e il numero minimo di istanze integre è impostato su 9, esegue la CodeDeploy distribuzione su 1 istanza alla volta.
- Se il gruppo di distribuzione ha 10 istanze e il numero minimo di istanze integre è impostato su 3, CodeDeploy esegue la distribuzione su 7 istanze contemporaneamente nel primo batch e quindi sulle 3 rimanenti nel secondo batch.
- Se il gruppo di distribuzione ha 10 istanze e il numero minimo di istanze integre è impostato su 0, esegue la CodeDeploy distribuzione su 10 istanze contemporaneamente.

Esempi

I seguenti esempi presuppongono un gruppo di distribuzione con 10 istanze.

Valore mimino di istanze integre: 95%

CodeDeploy arrotonda il numero minimo di istanze integre a 10 istanze, pari al numero di istanze integre. La distribuzione complessiva ha immediatamente esito negativo senza che la revisione venga distribuita ad alcuna istanza.

Valore mimino di istanze integre: 9

CodeDeploy distribuisce la revisione su un'istanza alla volta. Se la distribuzione su una delle istanze fallisce, fallisce CodeDeploy immediatamente l'implementazione complessiva perché il numero di istanze integre è uguale al numero minimo di istanze integre. Come eccezione a questa regola, se l'ultima istanza non riesce, la distribuzione ha comunque esito positivo.

CodeDeploy continua la distribuzione, un'istanza alla volta, fino a quando una distribuzione non fallisce o la distribuzione complessiva non viene completata. Se tutte e 10 le distribuzioni hanno esito positivo, il gruppo di distribuzione includerà 10 istanze integre.

Valore mimino di istanze integre: 8

CodeDeploy distribuisce la revisione su due istanze alla volta. Se due di queste distribuzioni falliscono, fallisce CodeDeploy immediatamente l'intera distribuzione. Come eccezione a questa regola, se l'ultima istanza è la seconda a non riuscire, la distribuzione complessiva avrà comunque esito positivo.

Valore mimino di istanze integre: 0

CodeDeploy distribuisce la revisione all'intero gruppo di distribuzione contemporaneamente. Almeno una distribuzione a un'istanza deve riuscire per far sì che la distribuzione complessiva abbia esito positivo. Se 0 istanze sono integre, la distribuzione ha esito negativo. Ciò è dovuto al requisito che, per contrassegnare una distribuzione complessiva come riuscita, almeno un'istanza deve essere integra al termine della distribuzione complessiva, anche se il valore minimo delle istanze integre è 0.

Circa il numero minimo di istanze integre per zona di disponibilità

> **Note**
>
> Questa sezione utilizza i termini istanza e host in modo intercambiabile per fare riferimento alle istanze Amazon EC2.

Se esegui la distribuzione su istanze in diverse zone di disponibilità, puoi facoltativamente abilitare la zonal configuration funzionalità, che consente la distribuzione in una zona di disponibilità CodeDeploy alla volta.

Quando questa funzionalità è abilitata, CodeDeploy farà in modo che il numero di host integri rimanga al di sopra dei valori «numero minimo di host integri per zona» e «numero minimo di host integri». Se il numero di host integri scende al di sotto di entrambi i valori, la distribuzione CodeDeploy non riesce in tutte le zone di disponibilità.

Per calcolare il numero di host su cui eseguire la distribuzione contemporaneamente, CodeDeploy utilizza sia il valore «numero minimo di host integri per zona» che il valore di «host integri minimi». CodeDeploy utilizzerà il minor numero di calcoli *[A]* e, dove e dove sono: `[B]` `[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]
```

Dopo aver determinato il numero di host su cui effettuare la distribuzione contemporaneamente, esegue la CodeDeploy distribuzione sugli host in batch composti da quel numero, una zona di disponibilità alla volta, con una pausa (o «tempo di cottura») opzionale tra le zone.

Esempio

Se la tua distribuzione è configurata in questo modo:

- `[total-hosts]` è `200`
- `[minimum-healthy-hosts]` è `160`
- `[total-hosts-per-AZ]` è `100`
- `[minimum-healthy-hosts-per-AZ]` è `50`

Allora...

- `[A]` = `200 - 160 = 40`
- `[B]` = `100 - 50 = 50`
- `40`è inferiore a `50`

Pertanto, CodeDeploy verrà distribuito agli 40 host contemporaneamente.

In questo scenario, la distribuzione si svolge come segue:

1. CodeDeploy si distribuisce nella prima zona di disponibilità:
   a. CodeDeploy viene distribuito sui primi 40 host.
   b. CodeDeploy viene distribuito sugli host successivi40.
   c. CodeDeploy viene distribuito sugli host 20 rimanenti.

   La distribuzione nella prima zona di disponibilità è ora completa.
2. (Facoltativo) CodeDeploy attende che la distribuzione nella prima zona «riprenda», come definito dall'impostazione Durata del monitoraggio o Aggiungi una durata del monitoraggio per la prima zona. Se non ci sono problemi, CodeDeploy continua.
3. CodeDeploy si distribuisce nella seconda zona di disponibilità:
   a. CodeDeploy viene distribuito sui primi 40 host.
   b. CodeDeploy viene distribuito sugli host successivi40.
   c. CodeDeploy viene distribuito sugli host 20 rimanenti.

   La distribuzione nella seconda e ultima zona di disponibilità è ora completa.

Per ulteriori informazioni sulla funzionalità di configurazione zonale e su come specificare il numero minimo di istanze integre per zona di disponibilità, vedere. zonal configuration

Utilizzo delle configurazioni di distribuzione in CodeDeploy

Una configurazione della distribuzione è un set di regole e di condizioni di successo e di errore utilizzate da CodeDeploy durante una distribuzione. Queste regole e condizioni sono diverse, a seconda che si esegua la distribuzione su una piattaforma di EC2/On-Premises elaborazione, una piattaforma di elaborazione AWS Lambda o una piattaforma di calcolo Amazon ECS.

Configurazioni di distribuzione su una piattaforma di elaborazione EC2/on-premises

Quando si esegue la distribuzione su una piattaforma di EC2/On-Premises elaborazione, la configurazione di distribuzione specifica, tramite l'uso del valore «numero minimo di host integri» e di un valore opzionale «numero minimo di host integri per zona», il numero o la percentuale di istanze che devono rimanere disponibili in qualsiasi momento durante un'implementazione.

È possibile utilizzare una delle tre configurazioni di distribuzione predefinite fornite da o creare una configurazione di distribuzione personalizzata. AWS Per ulteriori informazioni sulla creazione di configurazioni di distribuzione personalizzate, vedere. Create a Deployment Configuration
Se non si specifica una configurazione di distribuzione, CodeDeploy utilizza la configurazione CodeDeployDefault.OneAtATime di distribuzione.

Per ulteriori informazioni su come CodeDeploy monitora e valuta lo stato delle istanze durante una distribuzione, consulta. Instance Health Per visualizzare un elenco delle configurazioni di distribuzione già registrate sul tuo AWS account, consulta. View Deployment Configuration Details

Configurazioni di distribuzione predefinite per una piattaforma di elaborazione EC2/on-premises

La tabella seguente elenca le configurazioni della distribuzione predefinite.

> **Note**
>
> Non esistono configurazioni di distribuzione predefinite che supportino la zonal configuration funzionalità (ovvero la funzionalità che consente di specificare il numero di host integri per zona di disponibilità). Se si desidera utilizzare questa funzionalità, è necessario creare una configurazione di distribuzione personalizzata.

Configurazione dell'implementazione	Description
CodeDeployDefault.AllAtOnce	In-place distribuzioni: Tenta di distribuire una revisione di un'applic azione su quante più istanze possibile in una volta sola. Se la versione dell'applicazione viene distribuita su una o più delle istanze, lo stato della distribuzione complessiva viene visualizzato come Riuscito. Se la versione dell'applicazione non viene distribuita su nessuna delle istanze, lo stato della distribuz ione complessiva viene visualizzato come Non riuscito. Utilizzando un esempio di nove istanze, CodeDeployDefault.AllAtOnce tenta di eseguire la distribuzione su tutte e nove le istanze contemporaneamente. La distribuz ione complessiva ha esito positivo se la distribuzione di anche una singola istanza viene eseguita correttamente. Ha esito negativo solo se le distribuzioni a tutte e nove le istanze non riescono. Blue/green distribuzioni: • Distribuzione in un ambiente sostitutivo: segue le stesse regole di distribuzione delle distribuzioni sul CodeDeployDefault. AllAtOnce posto. • Reinstradamento del traffico: instrada il traffico verso tutte le istanze nell'ambiente di sostituzione tutto in una volta. Si conclude con successo se il traffico è correttam ente reinstradato verso almeno un'istanza. Fallisce se il reinstradamento si conclude con esito negativo per tutte le istanze.
CodeDeployDefault.HalfAtATime	In-place distribuzioni:

Configurazione dell'implementazione	Description
	Distribuisce su fino a metà delle istanze alla volta (con frazioni arrotondate per difetto). La distribuzione complessiva va a buon fine se la revisione dell'applicazione viene distribuita su almeno metà delle istanze (con frazioni arrotondate per eccesso). In caso contrario, la distribuzione ha esito negativo. Nell'esempio di nove istanze, effettua la distribuzione su fino a quattro istanze alla volta. La distribuzione complessiva va a buon fine se si conclude con esito positivo su cinque o più istanze. In caso contrario, la distribuzione ha esito negativo. **Note** Se esegui la distribuzione su istanze in più gruppi di Auto Scaling CodeDeploy , eseguirai la distribuzione su un massimo della metà delle istanze alla volta indipendentemente dal gruppo di Auto Scaling in cui si trovano. Ad esempio, supponiamo di avere due gruppi di Auto Scaling `ASG1` e `ASG2` ciascuno di 10 istanze. In questo scenario, CodeDeploy potrei implementare fino a 10 istanze in un colpo solo `ASG1` e considerarlo un successo perché è stato distribuito su almeno la metà delle istanze. Blue/green distribuzioni: • Distribuzione in un ambiente sostitutivo: segue le stesse regole di distribuzione

Configurazione dell'implementazione	Description
	delle distribuzioni sul CodeDeployDefault. HalfAtATime posto. • Reinstradamento del traffico: instrada il traffico verso fino alla metà delle istanze dell'ambiente di sostituzione alla volta. Riesce se il reinstradamento va a buon fine per almeno metà delle istanze. In caso contrario, produce un errore.

Configurazione dell'implementazione	Description
CodeDeployDefault.OneAtATime	In-place distribuzioni: Consente di distribuire la versione dell'appl icazione su una sola istanza alla volta. Per i gruppi di distribuzione che contengono più istanze: • La distribuzione complessiva va a buon fine se la versione dell'applicazione viene distribui ta su tutte le istanze. Come eccezione a questa regola, se la distribuzione non ha esito positivo sull'ultima istanza, la distribuz ione complessiva è comunque considera ta conclusa con successo. Questo perché CodeDeploy consente di portare offline solo un'istanza alla volta con la CodeDeplo yDefault.OneAtATime configurazione. • La distribuzione complessiva ha esito negativo non appena la versione dell'appl icazione non viene distribuita con successo su una qualsiasi delle istanze, tranne l'ultima. • In un esempio di nove istanze, effettua la distribuzione su un'istanza alla volta. La distribuzione complessiva ha esito positivo se la distribuzione alle prime otto istanze viene eseguita correttamente. La distribuzione complessiva ha esito negativo se la distribuz ione a una qualsiasi delle prime otto istanze non viene eseguita correttamente. Nel caso di gruppi di distribuzione che contengono solo un'istanza, la distribuzione complessiva va a buon fine solo se la distribuz ione sulla singola istanza va a buon fine.

Configurazione dell'implementazione	Description
	Blue/green distribuzioni: • Distribuzione in un ambiente sostitutivo: segue le stesse regole di distribuzione delle distribuzioni sul CodeDeployDefault. OneAtATime posto. • Reinstradamento del traffico: instrada il traffico verso un'istanze dell'ambiente di sostituzione alla volta. Va a buon fine se il traffico è correttamente reinstradato verso tutte le istanze sostitutive. Fallisce dopo il primo fallimento del reinstradamento. Come eccezione a questa regola, se l'ultima istanza non ha esito positivo, la distribuz ione complessiva è comunque considerata conclusa con successo.

Configurazioni delle implementazioni su una piattaforma di calcolo Amazon ECS

Quando esegui la distribuzione su una piattaforma di calcolo Amazon ECS, la configurazione di distribuzione specifica in che modo il traffico viene spostato verso il set di attività Amazon ECS aggiornato. Puoi spostare il traffico utilizzando una configurazione di distribuzione canaria, lineare o completa. Per ulteriori informazioni, consulta Configurazione dell'implementazione.

È anche possibile creare la propria configurazione della distribuzione di tipo canary o lineare. Per ulteriori informazioni, consulta Create a Deployment Configuration.

Configurazioni di distribuzione predefinite per una piattaforma di calcolo Amazon ECS

La tabella seguente elenca le configurazioni predefinite disponibili per le distribuzioni di Amazon ECS.

> **Note**
>
> Se utilizzi un Network Load Balancer, è supportata solo la configurazione di distribuzione `CodeDeployDefault.ECSAllAtOnce` predefinita.

Configurazione dell'implementazione	Description
CodeDeployDefault.ECSLinear10PercentEvery1Minutes	Reinstrada il 10% del traffico ogni minuto fino al reinstradamento di tutto il traffico.
CodeDeployDefault.ECSLinear10PercentEvery3Minutes	Reinstrada il 10% del traffico ogni tre minuti fino al reinstradamento di tutto il traffico.
CodeDeployDefault.ECSCanary10Percent5Minutes	Reinstrada il 10% del traffico nel primo blocco incrementale. Il restante 90% viene reinstradato cinque minuti più tardi.
CodeDeployDefault.ECSCanary10Percent15Minutes	Reinstrada il 10% del traffico nel primo blocco incrementale. Il restante 90 percento viene implementato 15 minuti dopo.
CodeDeployDefault.ECSAllAtOnce	Sposta tutto il traffico verso il container Amazon ECS aggiornato contemporaneamente.

Configurazioni di distribuzione per CloudFormation blue/green distribuzioni (Amazon ECS)

Quando esegui la distribuzione su una piattaforma di calcolo Amazon ECS tramite CloudFormation blue/green distribuzioni, la configurazione di distribuzione specifica in che modo il traffico viene spostato verso il contenitore Amazon ECS aggiornato. Puoi spostare il traffico utilizzando una configurazione di distribuzione canaria, lineare o completa. Per ulteriori informazioni, consulta Configurazione dell'implementazione.

Con le CloudFormation blue/green distribuzioni, non è possibile creare una configurazione di distribuzione canaria o lineare personalizzata. Per istruzioni dettagliate sull'utilizzo CloudFormation

per gestire le distribuzioni di Amazon ECS, consulta Automatizza le blue/green distribuzioni ECS tramite l'uso nella Guida per l'utente. blue/green CodeDeploy CloudFormationCloudFormation

> **Note**
>
> La gestione delle blue/green distribuzioni di Amazon ECS con non CloudFormation è disponibile nelle regioni di Europa (Milano), Africa (Città del Capo) e Asia Pacifico (Osaka).

Configurazioni di distribuzione su un AWS Lambda piattaforma di calcolo

Quando esegui la distribuzione su una piattaforma di elaborazione AWS Lambda, la configurazione di distribuzione specifica il modo in cui il traffico viene spostato verso le nuove versioni della funzione Lambda dell'applicazione. Puoi spostare il traffico utilizzando una configurazione di distribuzione canaria, lineare o completa. Per ulteriori informazioni, consulta Configurazione dell'implementazione.

È anche possibile creare la propria configurazione della distribuzione di tipo canary o lineare. Per ulteriori informazioni, consulta Create a Deployment Configuration.

Configurazioni di distribuzione predefinite per un AWS Lambda piattaforma di calcolo

La tabella seguente elenca le configurazioni della distribuzione predefinite disponibili per le distribuzioni su AWS Lambda .

Configurazione dell'implementazione	Description
CodeDeployDefault.LambdaCanary10Percent5Minutes	Reinstrada il 10% del traffico nel primo blocco incrementale. Il restante 90% viene reinstradato cinque minuti più tardi.
CodeDeployDefault.LambdaCanary10Percent10Minutes	Reinstrada il 10% del traffico nel primo blocco incrementale. Il restante 90% viene reinstradato 10 minuti più tardi.

Configurazione dell'implementazione	Description
CodeDeployDefault.LambdaCanary10Percent15Minutes	Reinstrada il 10% del traffico nel primo blocco incrementale. Il restante 90 percento viene implementato 15 minuti dopo.
CodeDeployDefault.LambdaCanary10Percent30Minutes	Reinstrada il 10% del traffico nel primo blocco incrementale. Il restante 90% viene reinstradato 30 minuti più tardi.
CodeDeployDefault.LambdaLinear10PercentEvery1Minute	Reinstrada il 10% del traffico ogni minuto fino al reinstradamento di tutto il traffico.
CodeDeployDefault.LambdaLinear10PercentEvery2Minutes	Reinstrada il 10% del traffico ogni due minuti fino al reinstradamento di tutto il traffico.
CodeDeployDefault.LambdaLinear10PercentEvery3Minutes	Reinstrada il 10% del traffico ogni tre minuti fino al reinstradamento di tutto il traffico.
CodeDeployDefault.LambdaLinear10PercentEvery10Minutes	Reinstrada il 10% del traffico ogni 10 minuti fino al reinstradamento di tutto il traffico.
CodeDeployDefault.LambdaAllAtOnce	Sposta tutto il traffico verso le funzioni Lambda aggiornate contemporaneamente.

Argomenti

- Create a Deployment Configuration
- View Deployment Configuration Details
- Delete a Deployment Configuration

Crea una configurazione di distribuzione con CodeDeploy

Se non desideri utilizzare una delle configurazioni di distribuzione predefinite fornite CodeDeploy, puoi crearne una personalizzata utilizzando le seguenti istruzioni.

Puoi utilizzare la CodeDeploy console AWS CLI, le CodeDeploy API o un CloudFormation modello per creare configurazioni di distribuzione personalizzate.

Per informazioni sull'utilizzo di un CloudFormation modello per creare una configurazione di distribuzione, consulta. CloudFormation modelli di CodeDeploy riferimento

Argomenti

- Creazione di una configurazione di distribuzione (console)
- Creazione di una configurazione di distribuzione con CodeDeploy (AWS CLI)

Creazione di una configurazione di distribuzione (console)

Utilizza le seguenti istruzioni per creare una configurazione di distribuzione utilizzando la AWS console.

Per creare una configurazione di distribuzione CodeDeploy utilizzando la console

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, scegli Configurazioni di distribuzione.

   Viene visualizzato un elenco di configurazioni di distribuzione integrate.
3. Selezionare Create deployment configuration (Crea configurazione della distribuzione).
4. In Nome di configurazione di distribuzione, immettere un nome per la configurazione di distribuzione. Ad esempio, **`my-deployment-config`**.
5. In Piattaforma Compute, scegli una delle seguenti opzioni:

   - EC2/On-premises
   - AWS Lambda
   - Amazon ECS
6. Esegui una delle seguenti operazioni:

   - Se hai scelto EC2/On-premises:

1. In Numero minimo di host integri, specifica il numero o la percentuale di istanze che devono rimanere disponibili in qualsiasi momento durante una distribuzione. Per ulteriori informazioni su come CodeDeploy monitora e valuta lo stato di salute di un'istanza durante una distribuzione, consulta Instance Health.
2. (Facoltativo) In Configurazione zonale, seleziona Abilita configurazione zonale per CodeDeploy distribuire l'applicazione in una zona di disponibilità alla volta, all'interno di una regione. AWS Effettuando l'implementazione in una zona di disponibilità alla volta, è possibile esporre l'implementazione a un pubblico sempre più vasto man mano che aumenta la fiducia nelle prestazioni e nella fattibilità dell'implementazione. Se non abiliti una configurazione zonale, CodeDeploy distribuisce l'applicazione su una selezione casuale di host in una regione.

   Se abiliti la funzionalità di configurazione zonale, tieni presente quanto segue:

   - La funzionalità di configurazione zonale è supportata solo con distribuzioni in loco su istanze Amazon EC2. (le Blue/green distribuzioni e le istanze locali non sono supportate.) Per ulteriori informazioni sulle distribuzioni sul posto, consulta. Il tipo di implementazione
   - La funzionalità di configurazione zonale non è supportata con configurazioni di distribuzione predefinite. Per utilizzare una configurazione zonale, è necessario creare una configurazione di distribuzione personalizzata, come descritto qui.
   - Se è CodeDeploy necessario ripristinare una distribuzione, CodeDeploy eseguirà le operazioni di rollback su host casuali. (non CodeDeploy eseguirà il rollback di una zona alla volta, come ci si potrebbe aspettare.) Questo comportamento di rollback è stato scelto per motivi di prestazioni. Per ulteriori informazioni sui rollback, vedere. Ridistribuisci e ripristina una distribuzione con CodeDeploy
3. Se avete selezionato la casella di controllo Abilita configurazione zonale, specificate facoltativamente le seguenti opzioni:

   - (Facoltativo) In Durata del monitoraggio, specificare il periodo di tempo, in secondi, che CodeDeploy deve trascorrere dopo il completamento di una distribuzione in una zona di disponibilità. CodeDeploy aspetterà questo periodo di tempo prima di iniziare una distribuzione nella successiva zona di disponibilità. Prendi in considerazione l'aggiunta di una durata del monitoraggio per dare alla distribuzione un po' di tempo per mettersi alla prova (o «funzionare») in una zona di disponibilità prima di essere rilasciata nella zona successiva. Se non specifichi la durata del monitoraggio, CodeDeploy inizia immediatamente la distribuzione nella zona di disponibilità successiva. Per

ulteriori informazioni su come funziona l'impostazione della durata del monitoraggio, consultaCirca il numero minimo di istanze integre per zona di disponibilità.

- (Facoltativo) Seleziona Aggiungi una durata del monitoraggio per la prima zona per impostare una durata del monitoraggio che si applica solo alla prima zona di disponibilità. È possibile impostare questa opzione se si desidera consentire tempi di cottura aggiuntivi per la prima zona di disponibilità. Se non specifichi un valore in Aggiungi la durata del monitoraggio della prima zona, CodeDeploy utilizza il valore della durata del monitoraggio per la prima zona di disponibilità.
- (Facoltativo) In Numero minimo di host integri per zona, specifica il numero o la percentuale di istanze che devono rimanere disponibili per zona di disponibilità durante una distribuzione. Scegliete FLEET_PERCENT per specificare una percentuale o HOST_COUNT per specificare un numero. Questo campo funziona in combinazione con il campo Minimum healthy hosts. Per ulteriori informazioni, consulta Circa il numero minimo di istanze integre per zona di disponibilità.

  Se non specifichi un valore in Minimum healthy hosts per zone, CodeDeploy utilizza un valore predefinito `0` pari alla percentuale.

- Se hai scelto AWS LambdaAmazon ECS:

  1. Per Tipo, scegli Linear o Canary.
  2. Nei campi Fase e Intervallo, effettuate una delle seguenti operazioni:

     - Se hai scelto Canary, per Step, inserisci una percentuale di traffico, compresa tra 1 e 99, da spostare. Questa è la percentuale di traffico che viene spostata nel primo incremento. Il traffico rimanente viene reinstradato con il secondo blocco incrementale dopo l'intervallo di tempo selezionato.

       Per Intervallo, inserisci il numero di minuti tra il primo e il secondo cambio di traffico.

     - Se hai scelto Linear, per Step, inserisci una percentuale di traffico, compresa tra 1 e 99, da spostare. Questa è la percentuale di traffico che viene spostata all'inizio di ogni intervallo.

       Per Intervallo, inserisci il numero di minuti tra ogni spostamento incrementale.

7. Selezionare Create deployment configuration (Crea configurazione della distribuzione).

Ora hai una configurazione di distribuzione che puoi associare a un gruppo di distribuzione.

Creazione di una configurazione di distribuzione con CodeDeploy (AWS CLI)

Per utilizzare la AWS CLI per creare una configurazione di distribuzione, chiamate il comando create-deployment-config.

L'esempio seguente crea una configurazione di EC2/On-Premises distribuzione denominata `ThreeQuartersHealthy` che richiede che il 75% delle istanze di destinazione rimanga integro durante una distribuzione:

```
aws deploy create-deployment-config --deployment-config-name ThreeQuartersHealthy --
minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

L'esempio seguente crea una configurazione di EC2/On-Premises distribuzione denominata `300Total50PerAZ` che richiede che 300 istanze di destinazione rimangano integre in totale per distribuzione e 50 per zona di disponibilità. Imposta inoltre una durata del monitoraggio di 1 ora.

```
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}}'
```

L'esempio seguente crea una configurazione di distribuzione AWS Lambda denominata. `Canary25Percent45Minutes` Usa lo spostamento del traffico Canary per spostare il 25% del traffico nel primo incremento. Il restante 75% viene spostato 45 minuti più tardi.

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
 --traffic-routing-config
 "type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform Lambda
```

L'esempio seguente crea una configurazione di distribuzione Amazon ECS denominata`Canary25Percent45Minutes`. Usa lo spostamento del traffico Canary per spostare il 25% del traffico nel primo incremento. Il restante 75% viene spostato 45 minuti più tardi.

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
 --traffic-routing-config
 "type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform ECS
```

Visualizza i dettagli della configurazione di distribuzione con CodeDeploy

Puoi utilizzare la CodeDeploy console, le o le AWS CLI CodeDeploy API per visualizzare i dettagli sulle configurazioni di distribuzione associate al tuo account. AWS Per le descrizioni delle configurazioni di CodeDeploy distribuzione predefinite, vedere. Configurazioni di distribuzione predefinite per una piattaforma di elaborazione EC2/on-premises

Argomenti

- Visualizza i dettagli della configurazione di distribuzione (console)
- Visualizza la configurazione di distribuzione (CLI)

Visualizza i dettagli della configurazione di distribuzione (console)

Per utilizzare la CodeDeploy console per visualizzare un elenco di nomi di configurazione di distribuzione:

1. Accedi Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy e scegli Configurazioni di distribuzione.

   Di seguito puoi visualizzare i nomi delle configurazioni della distribuzione e i criteri associati a ciascuna configurazione della distribuzione.

> **Note**
>
> Se non viene visualizzata alcuna voce, assicurati che sia selezionata la regione corretta. Nella barra di navigazione, nel selettore di regione, scegli una delle regioni elencate in Regione ed endpoint in. Riferimenti generali di AWS CodeDeploy è supportato solo in queste regioni.

Visualizza la configurazione di distribuzione (CLI)

Per utilizzare il AWS CLI per visualizzare i dettagli della configurazione di distribuzione, chiama il `get-deployment-config` comando o il `list-deployment-configs` comando.

Per visualizzare i dettagli su una singola configurazione di distribuzione, chiama il comando get-deployment-config, specificando il nome univoco della configurazione di distribuzione.

Per visualizzare i dettagli sulle configurazioni di distribuzione multiple, chiama il comando list-deployments.

Eliminare una configurazione di distribuzione con CodeDeploy

Puoi utilizzare le API AWS CLI o le CodeDeploy API per eliminare le configurazioni di distribuzione personalizzate associate al tuo AWS account. Non è possibile eliminare le configurazioni della distribuzione integrate come `CodeDeployDefault.AllAtOnce`, `CodeDeployDefault.HalfAtATime` e `CodeDeployDefault.OneAtATime`.

> **Warning**
>
> Non è possibile eliminare una configurazione della distribuzione personalizzata ancora in uso. Se viene eliminata una configurazione della distribuzione personalizzata non utilizzata, non sarà più possibile associarla con nuove distribuzioni e con nuovi gruppi di distribuzione. Questa operazione non può essere annullata.

Per utilizzare per AWS CLI eliminare una configurazione di distribuzione, chiama il comando delete-deployment-config, specificando il nome della configurazione di distribuzione. Per visualizzare un elenco di nomi di configurazione di distribuzione, chiama il comando list-deployment-configs.

L'esempio seguente elimina una configurazione di distribuzione denominata. ThreeQuartersHealthy

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Utilizzo delle applicazioni in CodeDeploy

Dopo aver configurato istanze, ma prima di distribuire una revisione, è necessario creare un'applicazione in CodeDeploy. Un'applicazione è semplicemente un nome o un container utilizzato da CodeDeploy per garantire che, durante una distribuzione, si faccia riferimento alla revisione, alla configurazione di distribuzione e al gruppo di distribuzione corretti.

Utilizza le informazioni nella tabella seguente per le fasi successive:

Compute platform (Piattaforma di calcolo)	Scenario	Informazioni per la fase successiva
EC2/On-Premises	Non ho ancora creato istanze.	Consulta Utilizzo delle istanze per CodeDeploy, quindi torna a questa pagina.
EC2/On-Premises	Ho create istanze, ma non ho completato il tagging.	Consulta Tagging Instances for Deployments, quindi torna a questa pagina.
EC2/On-Premises, AWS Lambda e Amazon ECS	Non ho ancora creato un'applicazione.	Per informazioni, consultar e Crea un'applicazione con CodeDeploy.
EC2/On-Premises, AWS Lambda e Amazon ECS	Ho già creato un'applicazione, ma non ho creato un gruppo di distribuzione.	Per informazioni, consulta Crea un gruppo di distribuz ione con CodeDeploy.
EC2/On-Premises, AWS Lambda e Amazon ECS	Ho già creato un'applicazione e un gruppo di distribuz ione, ma non ho creato una revisione dell'applicazione.	Per informazioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.
EC2/On-Premises, AWS Lambda e Amazon ECS	Ho già creato un'applicazione e un gruppo di distribuzione e ho già caricato la revisione dell'applicazione. Sono pronto per la distribuzione.	Per informazioni, consulta Crea una distribuzione con CodeDeploy.

Argomenti

- Crea un'applicazione con CodeDeploy
- Visualizza i dettagli dell'applicazione con CodeDeploy
- Creazione di una regola di notifica
- Rinominare un'applicazione CodeDeploy
- Eliminare un'applicazione in CodeDeploy

Crea un'applicazione con CodeDeploy

Un'applicazione è semplicemente un nome o un contenitore utilizzato da CodeDeploy per garantire che durante una distribuzione venga fatto riferimento alla revisione, alla configurazione di distribuzione e al gruppo di distribuzione corretti. È possibile utilizzare la CodeDeploy console AWS CLI, le CodeDeploy API o un CloudFormation modello per creare applicazioni.

Il codice, o la revisione dell'applicazione, viene installato nelle istanze tramite un processo chiamato distribuzione. CodeDeploy supporta due tipi di implementazioni:

- In-place distribuzione: l'applicazione su ogni istanza del gruppo di distribuzione viene interrotta, viene installata la revisione più recente dell'applicazione e la nuova versione dell'applicazione viene avviata e convalidata. È possibile utilizzare un sistema di bilanciamento del carico in modo che ogni istanza venga annullata durante la distribuzione e quindi ripristinata in servizio al termine della distribuzione. Solo le distribuzioni che utilizzano la piattaforma di EC2/On-Premises elaborazione possono utilizzare le distribuzioni sul posto. Per ulteriori informazioni sulle implementazioni sul posto, consulta. Panoramica di una distribuzione sul posto
- Blue/green implementazione: il comportamento della distribuzione dipende dalla piattaforma di elaborazione utilizzata:
  - Blue/green su una piattaforma di EC2/On-Premises elaborazione: le istanze in un gruppo di distribuzione (l'ambiente originale) vengono sostituite da un diverso set di istanze (l'ambiente sostitutivo) utilizzando questi passaggi:
    - Le istanze vengono fornite per l'ambiente sostitutivo.
    - La revisione più recente dell'applicazione viene installata sulle istanze sostitutive.
    - Si verifica un tempo di attesa opzionale per attività come il test delle applicazioni e la verifica del sistema.
    - Le istanze nell'ambiente sostitutivo vengono registrate con uno o più sistemi di bilanciamento del carico Elastic Load Balancing, causando il reindirizzamento del traffico verso di essi. Le

istanze nell'ambiente originale vengono cancellate e possono essere terminate o mantenute in esecuzione per altri usi.

> **Note**
>
> Se utilizzi una piattaforma di EC2/On-Premises elaborazione, tieni presente che le blue/green distribuzioni funzionano solo con istanze Amazon EC2.

- Blue/green su una piattaforma AWS Lambda di calcolo Amazon ECS: il traffico viene spostato in incrementi in base a una configurazione di distribuzione canaria, lineare o completa.
- Blue/green implementazioni tramite CloudFormation: Il traffico viene spostato dalle risorse correnti alle risorse aggiornate come parte di un aggiornamento dello stack. CloudFormation Attualmente sono supportate solo le implementazioni ECS blue/green .

Per ulteriori informazioni sulle blue/green distribuzioni, consulta. Panoramica di una distribuzione blue/green

Quando si utilizza la CodeDeploy console per creare un'applicazione, si configura contemporaneamente il primo gruppo di distribuzione. Quando si utilizza il AWS CLI per creare un'applicazione, si crea il primo gruppo di distribuzione in un passaggio separato.

Per visualizzare un elenco di applicazioni già registrate sul tuo AWS account, consultaVisualizza i dettagli dell'applicazione con CodeDeploy. Per informazioni sull'utilizzo di un CloudFormation modello per creare un'applicazione, vedereCloudFormation modelli di CodeDeploy riferimento.

Entrambi i tipi di distribuzione non si applicano a tutte le destinazioni. La tabella seguente elenca quali tipi di distribuzione sono compatibili rispetto alle distribuzioni verso tre tipi di destinazioni della distribuzione.

Destinazione della distribuzione	In-place	Blue/green
Amazon EC2	Sì	Sì
On-premises	Sì	No
Funzioni AWS Lambda serverless	No	Sì

Destinazione della distribuzione	In-place	Blue/green
Applicazioni Amazon ECS	No	Sì

Argomenti

- Crea un'applicazione per un'implementazione sul posto (console)
- Crea un'applicazione per una blue/green distribuzione (console)
- Crea un'applicazione per l'implementazione di un servizio Amazon ECS (console)
- Crea un'applicazione per un AWS Lambda implementazione della funzione (console)
- Creare un'applicazione (CLI)

Crea un'applicazione per un'implementazione sul posto (console)

Per utilizzare la CodeDeploy console per creare un'applicazione per un'implementazione sul posto:

> **Warning**
>
> Non seguire i passaggi seguenti se:
>
> - Non hai preparato le istanze per l'uso nelle CodeDeploy distribuzioni. Per configurare le istanze, segui le istruzioni che trovi su Utilizzo delle istanze per CodeDeploy e quindi segui i passaggi descritti in questa sezione.
> - Desideri creare un'applicazione che utilizza una configurazione della distribuzione personalizzata, ma non hai ancora creato la configurazione della distribuzione. Segui le istruzioni che trovi su Create a Deployment Configuration e quindi segui i passaggi descritti in questa sezione.
> - Non disponi di un ruolo di servizio affidabile CodeDeploy con la fiducia e le autorizzazioni minime richieste. Per creare e configurare un ruolo di servizio con le autorizzazioni necessarie, segui le istruzioni contenute in Fase 2: Creare un ruolo di servizio per CodeDeploy e quindi torna alla procedura descritta in questa sezione.

- Vuoi selezionare un Classic Load Balancer, Application Load Balancer o Network Load Balancer in Elastic Load Balancing per la distribuzione sul posto, ma non l'hai ancora creato.

Per creare un'applicazione per una distribuzione sul posto utilizzando la console: CodeDeploy

1. Accedi Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandere Deploy (Distribuisci) e scegliere Getting started (Nozioni di base).
3. Scegli Crea applicazione.
4. Specificare il nome dell´applicazione nel campo Application name (Nome applicazione).
5. Da Compute Platform, scegli EC2/On-premises.
6. Scegli Crea applicazione.
7. Nella pagina delle applicazioni, nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione).
8. In Deployment group name (Nome gruppo di distribuzione), inserire un nome che descriva il gruppo di distribuzione.

   > **Note**
   >
   > Se desideri utilizzare le stesse impostazioni utilizzate in un altro gruppo di distribuzione (incluso il nome del gruppo di distribuzione, i tag, i nomi dei gruppi di Amazon EC2 Auto Scaling o entrambi e la configurazione di distribuzione), specifica tali impostazioni in questa pagina. Sebbene questo nuovo gruppo di distribuzione e il gruppo di distribuzione esistente abbiano lo stesso nome, li CodeDeploy tratta come gruppi di distribuzione separati, poiché ciascuno di essi è associato ad applicazioni separate.

9. In Service role (Ruolo di servizio), scegliere un ruolo di servizio che conceda a CodeDeploy l'accesso all'istanza di destinazione.

10. In Tipo di distribuzione, scegli In-place.
11. In Environment configuration (Configurazione dell'ambiente), scegliere una delle seguenti opzioni:

    a. Gruppi Amazon EC2 Auto Scaling: inserisci o scegli il nome di un gruppo Amazon EC2 Auto Scaling su cui distribuire la revisione dell'applicazione. Quando vengono lanciate nuove istanze Amazon EC2 come parte di un CodeDeploy gruppo Amazon EC2 Auto Scaling, puoi distribuire automaticamente le tue revisioni alle nuove istanze. Puoi aggiungere fino a 10 gruppi Amazon EC2 Auto Scaling a un gruppo di distribuzione.

    b. Istanze o On-premisesistanze Amazon EC2: nei campi Chiave e Valore, inserisci i valori della coppia chiave-valore che hai usato per etichettare le istanze. A un singolo gruppo di tag puoi aggiungere fino a 10 coppie chiave-valore associate ai tag.

        i. Puoi utilizzare i caratteri jolly nel campo Valore per identificare tutte le istanze etichettate secondo determinati schemi, ad esempio nomi di istanze Amazon EC2, centri di costo e gruppi simili e così via. Ad esempio, se scegli Nome nel campo Chiave e inserisci **`GRP-*a`** nel campo Valore, CodeDeploy identifica tutte le istanze che corrispondono a quel modello, ad esempio, e. **`GRP-1a GRP-2a GRP-XYZ-a`**

        ii. Il campo Value (Valore) prevede la distinzione tra lettere maiuscole e minuscole.

        iii. Per rimuovere una coppia chiave-valore dall'elenco, selezionare Remove tag (Rimuovi tag).

    Quando CodeDeploy trova le istanze che corrispondono a ciascuna coppia chiave-valore specificata o al nome del gruppo Amazon EC2 Auto Scaling, visualizza il numero di istanze corrispondenti. Selezionare il numero per visualizzare ulteriori informazioni sulle istanze.

    Se desideri rifinire i criteri per le istanze distribuite, scegli Aggiungi gruppo di tag per creare un gruppo di tag. È possibile creare fino a tre gruppi di tag contenenti ciascuno fino a dieci coppie chiave-valore. Quando si utilizzano più gruppi di tag in un gruppo di distribuzione, sono incluse nel gruppo di distribuzione solo le istanze che sono identificate da tutti i gruppi di tag. Ciò significa che un'istanza, per essere inclusa nel gruppo di distribuzione, deve essere associata ad almeno uno dei tag in ciascuno dei gruppi.

    Per informazioni su come usare i gruppi di tag per raffinare il gruppo di distribuzione, consulta Tagging Instances for Deployments.

12. In Deployment configuration (Configurazione della distribuzione) scegliere una configurazione della distribuzione per controllare la velocità di distribuzione dell´applicazione sulle istanze, ad esempio una per volta o tutte insieme. Per ulteriori dettagli sulle configurazioni della distribuzione, consulta Utilizzo delle configurazioni di distribuzione in CodeDeploy.
13. (Facoltativo) In Load balancer, seleziona Enable load balancing, quindi dagli elenchi seleziona i gruppi target Classic Load Balancer, Application Load Balancer e Network Load Balancer per gestire il traffico verso le istanze durante la distribuzione. CodeDeploy È possibile selezionare fino a 10 Classic Load Balancer e 10 gruppi target, per un totale di 20 articoli. Assicurati che le istanze Amazon EC2 su cui desideri effettuare la distribuzione siano registrate con i sistemi di bilanciamento del carico (Classic Load Balancer) o i gruppi target (Application Load Balancer e Network Load Balancer) selezionati.

    Durante una distribuzione, le istanze originali vengono cancellate dai sistemi di bilanciamento del carico e dai gruppi target selezionati per evitare che il traffico venga indirizzato verso queste istanze durante la distribuzione. Una volta completata la distribuzione, ogni istanza viene registrata nuovamente con tutti i Classic Load Balancer e i gruppi target selezionati.

    Per ulteriori informazioni sui sistemi di bilanciamento del carico per le distribuzioni, consulta. CodeDeploy Integrating CodeDeploy with Elastic Load Balancing
14. (Facoltativo) Espandi Advanced e configura tutte le opzioni che desideri includere nella distribuzione, come i trigger di notifica di Amazon SNS, gli CloudWatch allarmi Amazon o i rollback automatici.

    Per ulteriori informazioni, consulta Configurazione delle opzioni avanzate per un gruppo di distribuzione.
15. Scegliere Create deployment group (Crea gruppo di distribuzione).

Il prossimo passaggio server per preparare la distribuzione di una versione rispetto all'applicazione e al gruppo di distribuzione. Per istruzioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.

Crea un'applicazione per una blue/green distribuzione (console)

Per utilizzare la CodeDeploy console per creare un'applicazione per una blue/green distribuzione:

> **Note**
>
> Una distribuzione sulla piattaforma di elaborazione AWS Lambda è sempre una distribuzione. blue/green Non si specifica un'opzione di tipo di distribuzione.

> **Warning**
>
> Non seguire i passaggi seguenti se:
>
> - Non ci sono istanze con l' CodeDeploy agente installato da sostituire durante il processo di blue/green distribuzione. Per configurare le istanze, segui le istruzioni che trovi su Utilizzo delle istanze per CodeDeploy e quindi segui i passaggi descritti in questa sezione.
> - Desideri creare un'applicazione che utilizza una configurazione della distribuzione personalizzata, ma non hai ancora creato la configurazione della distribuzione. Segui le istruzioni che trovi su Create a Deployment Configuration e quindi segui i passaggi descritti in questa sezione.
> - Non disponete di un ruolo di servizio che corrisponda CodeDeploy almeno alla fiducia e alle autorizzazioni descritte in. Fase 2: Creare un ruolo di servizio per CodeDeploy Per creare e configurare un ruolo di servizio, segui le istruzioni contenute in Fase 2: Creare un ruolo di servizio per CodeDeploy e quindi segui la procedura descritta in questa sezione.
> - Non hai creato un Classic Load Balancer, Application Load Balancer o Network Load Balancer in Elastic Load Balancing per la registrazione delle istanze nell'ambiente sostitutivo. Per ulteriori informazioni, consulta Configura un sistema di bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2.

1. Accedi a Console di gestione AWS e apri la console all'indirizzo. CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandere Deploy (Distribuisci) e scegliere Getting started (Nozioni di base).
3. Specificare il nome dell´applicazione nel campo Application name (Nome applicazione).

4. Da Compute platform (Piattaforma di elaborazione), scegliere EC2/On-Premises.
5. Scegli Crea applicazione.
6. Nella pagina delle applicazioni, nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione).
7. In Deployment group name (Nome gruppo di distribuzione), inserire un nome che descriva il gruppo di distribuzione.

> **Note**
>
> Se desideri utilizzare le stesse impostazioni utilizzate in un altro gruppo di distribuzione (inclusi i tag dei nomi dei gruppi di distribuzione, i nomi dei gruppi di Amazon EC2 Auto Scaling e la configurazione di distribuzione), scegli tali impostazioni in questa pagina. Anche se questo nuovo gruppo di distribuzione e il gruppo di distribuzione esistente possiedono lo stesso nome, CodeDeploy li tratta come gruppi di distribuzione separati, in quanto ognuno è associato a un'applicazione distinta.

8. In Service role (Ruolo di servizio), scegliere un ruolo di servizio che conceda a CodeDeploy l'accesso all'istanza di destinazione.
9. In Tipo di distribuzione scegli. Blue/green
10. In Environment configuration (Configurazione dell'ambiente), scegliere il metodo da utilizzare per fornire le istanze per l'ambiente di sostituzione:

    a. Copia automaticamente il gruppo Amazon EC2 Auto Scaling: crea un CodeDeploy gruppo Amazon EC2 Auto Scaling copiandone uno specificato.

    b. Manually provision instances (Provisioning manuale delle istanze): non è necessario specificare le istanze per l'ambiente di sostituzione finché non si crea una distribuzione. È necessario creare le istanze prima di avviare la distribuzione. Al contrario, qui è necessario specificare le istanze che si desidera sostituire.

11. A seconda dell'opzione scelta nella fase 10, eseguire una delle seguenti operazioni:

    - Se hai scelto Copia automaticamente il gruppo Amazon EC2 Auto Scaling: Nel gruppo Amazon EC2 Auto Scaling, scegli o inserisci il nome del gruppo Amazon EC2 Auto Scaling che desideri utilizzare come modello per il gruppo Amazon EC2 Auto Scaling per le istanze nell'ambiente sostitutivo. Il numero di istanze attualmente integre nel gruppo Amazon EC2 Auto Scaling scelto viene creato nell'ambiente sostitutivo.

- Se hai scelto il provisioning manuale delle istanze: abilita i gruppi Amazon EC2 Auto Scaling, le istanze Amazon EC2 o entrambi per specificare le istanze da aggiungere a questo gruppo di distribuzione. Inserisci i valori dei tag Amazon EC2 o i nomi dei gruppi Amazon EC2 Auto Scaling per identificare le istanze nel tuo ambiente originale (ovvero le istanze che desideri sostituire o che eseguono la revisione corrente dell'applicazione).

12. In Load balancer, seleziona Enable load balancing, quindi dagli elenchi seleziona i gruppi target Classic Load Balancer, Application Load Balancer e Network Load Balancer con cui desideri registrare le istanze Amazon EC2 sostitutive. Ogni istanza sostitutiva verrà registrata con tutti i Classic Load Balancer e i gruppi target selezionati. È possibile selezionare fino a 10 Classic Load Balancer e 10 gruppi target, per un totale di 20 articoli.

    Il traffico verrà reindirizzato dalle istanze originali a quelle sostitutive in base alle impostazioni di configurazione scelte per il reindirizzamento del traffico e la distribuzione.

    Per ulteriori informazioni sui sistemi di bilanciamento del carico per le distribuzioni, consulta. CodeDeploy Integrating CodeDeploy with Elastic Load Balancing
13. In Deployment settings (Impostazioni Distribuzione), esaminare le opzioni predefinite relative al reinstradamento del traffico verso l'ambiente di sostituzione, alla configurazione della distribuzione da usare per la distribuzione e alla modalità di gestione delle istanze dell'ambiente originale dopo la distribuzione.

    Per modificare le impostazioni, andare alla fase successiva. Altrimenti, passare alla fase 15.
14. Per modificare le impostazioni di distribuzione per la blue/green distribuzione, modifica una delle seguenti impostazioni.

Impostazione	Opzioni
Traffic rerouting (Reinstradamento del traffico)	• Reindirizza immediatamente il traffico: non appena le istanze nell'ambiente sostitutivo vengono fornite e su di esse viene installat a la revisione più recente dell'applicazione, queste vengono registrate automatic amente presso i sistemi di bilanciamento del carico e i gruppi target specificati, provocando il reindirizzamento del traffico verso di essi. Le istanze nell'ambiente originale sono quindi rimosse.

Impostazione	Opzioni
	• Sceglierò se reindirizzare il traffico: le istanze nell'ambiente sostitutivo non vengono registrate con i sistemi di bilanciamento del carico e i gruppi target specificati, a meno che non si reindiriz zino manualmente il traffico. Se il tempo di attesa indicato trascorre senza che il traffico sia stato reinstradato, lo stato della distribuzione si modifica in Arrestata.
Configurazione dell'implementazione	Scegli la velocità con cui le istanze nell'ambi ente sostitutivo vengono registrate presso i sistemi di bilanciamento del carico e i gruppi target, ad esempio una alla volta o tutte contemporaneamente. **Note** Dopo che il traffico è stato instradat o correttamente verso l'ambiente di sostituzione, le istanze nell'ambi ente originale sono rimosse tutte contemporaneamente indipende ntemente dalla configurazione della distribuzione scelta. Per ulteriori informazioni, consulta Utilizzo delle configurazioni di distribuzione in CodeDeploy.

Impostazione	Opzioni
Istanze originali	• Termina le istanze originali nel gruppo di distribuzione: dopo che il traffico è stato reindirizzato all'ambiente sostitutivo, le istanze che erano state cancellate dai sistemi di bilanciamento del carico e dai gruppi target vengono terminate dopo il periodo di attesa specificato. • Mantieni in esecuzione le istanze originali del gruppo di distribuzione: dopo che il traffico è stato reindirizzato all'ambie nte sostitutivo, le istanze che sono state cancellate dai sistemi di bilanciamento del carico e dai gruppi target vengono mantenute in esecuzione.

15. (Facoltativo) In Advanced, configura le opzioni che desideri includere nella distribuzione, come i trigger di notifica di Amazon SNS, gli CloudWatch allarmi Amazon o i rollback automatici.

    Per informazioni su come specificare le opzioni avanzate nei gruppi di distribuzione di gruppi, consulta Configurazione delle opzioni avanzate per un gruppo di distribuzione.

16. Scegliere Create deployment group (Crea gruppo di distribuzione).

Il prossimo passaggio server per preparare la distribuzione di una versione rispetto all'applicazione e al gruppo di distribuzione. Per istruzioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.

Crea un'applicazione per l'implementazione di un servizio Amazon ECS (console)

Puoi utilizzare la CodeDeploy console per creare un'applicazione per la distribuzione di un servizio Amazon ECS.

1. Accedi Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy e scegli Getting started.
3. Nella pagina Crea applicazione, scegli Usa CodeDeploy.
4. Specificare il nome dell´applicazione nel campo Application name (Nome applicazione).
5. Dalla piattaforma Compute, scegli Amazon ECS.
6. Scegli Crea applicazione.
7. Nella pagina delle applicazioni, nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione). Per ulteriori informazioni su ciò di cui hai bisogno per creare un gruppo di distribuzione per una distribuzione Amazon ECS, consultaPrima di iniziare una distribuzione Amazon ECS.
8. In Deployment group name (Nome gruppo di distribuzione), inserire un nome che descriva il gruppo di distribuzione.

> **Note**
>
> Per utilizzare le stesse impostazioni utilizzate in un altro gruppo di distribuzione (tra cui il nome del gruppo di distribuzione e la configurazione della distribuzione), scegliere tali impostazioni in questa pagina. Sebbene questo nuovo gruppo e il gruppo esistente possano avere lo stesso nome, li CodeDeploy considera come gruppi di distribuzione separati, poiché ciascuno è associato a un'applicazione separata.

9. In Ruolo di servizio, scegli un ruolo di servizio che garantisca CodeDeploy l'accesso ad Amazon ECS. Per ulteriori informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.
10. Da Load balancer name, scegli il nome del load balancer che fornisce il traffico al tuo servizio Amazon ECS.
11. Da Production listener port, scegli la porta e il protocollo per il listener che invia il traffico di produzione al tuo servizio Amazon ECS.
12. (Facoltativo) Da Test listener port, scegli la porta e il protocollo di un listener di test che serve il traffico verso l'attività sostitutiva impostata nel tuo servizio Amazon ECS durante la distribuzione. È possibile specificare una o più funzioni Lambda nel AppSpec file che viene eseguito durante l'hook. `AfterAllowTestTraffic` Le funzioni possono eseguire test di convalida. Se un test di convalida fallisce, viene attivato un rollback della distribuzione. Se i test di convalida riescono,

viene attivato l'hook successivo nel ciclo di vita della distribuzione, `BeforeAllowTraffic`. Se non viene specificata una porta per il listener di test, non succede nulla durante l'hook. `AfterAllowTestTraffic` Per ulteriori informazioni, consulta AppSpec sezione 'hook' per una distribuzione Amazon ECS.

13. Da Nome del gruppo di destinazione 1 e Nome del gruppo di destinazione 2, scegli i gruppi target utilizzati per instradare il traffico durante la distribuzione. CodeDeploy associa un gruppo target al set di attività originale del servizio Amazon ECS e l'altro al set di attività sostitutivo. Per ulteriori informazioni, consulta Target Groups for Your Application Load Balancers.
14. Scegli Reindirizzare immediatamente il traffico o Specificare quando reindirizzare il traffico per determinare quando reindirizzare il traffico verso il servizio Amazon ECS aggiornato.

    Se scegli Reroute il traffico immediatamente, la distribuzione reindirizza automaticamente il traffico dopo il provisioning del set di attività sostitutivo.

    Se scegli Specificare quando reindirizzare il traffico, scegli il numero di giorni, ore e minuti di attesa dopo il corretto provisioning del set di attività sostitutivo. Durante questo periodo di attesa, vengono eseguiti i test di convalida nelle funzioni Lambda specificate AppSpec nel file. Se il tempo di attesa scade prima che il traffico venga reindirizzato, lo stato di distribuzione cambia in. `Stopped`
15. Per la cessazione della revisione originale, scegli il numero di giorni, ore e minuti da attendere dopo una corretta distribuzione prima che l'attività originale impostata nel tuo servizio Amazon ECS venga interrotta.
16. (Facoltativo) In Advanced, configura tutte le opzioni che desideri includere nella distribuzione, come i trigger di notifica di Amazon SNS, gli CloudWatch allarmi Amazon o i rollback automatici.

    Per ulteriori informazioni, consulta Configurazione delle opzioni avanzate per un gruppo di distribuzione.

Crea un'applicazione per un AWS Lambda implementazione della funzione (console)

È possibile utilizzare la CodeDeploy console per creare un'applicazione per l'implementazione di una AWS Lambda funzione.

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy e scegli Getting started.
3. Nella pagina Crea applicazione, scegli Usa CodeDeploy.
4. Specificare il nome dell'applicazione nel campo Application name (Nome applicazione).
5. Da Compute platform (Piattaforma di elaborazione), scegliere AWS Lambda.
6. Scegli Crea applicazione.
7. Nella pagina delle applicazioni, nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione).
8. In Deployment group name (Nome gruppo di distribuzione), inserire un nome che descriva il gruppo di distribuzione.

   > **Note**
   >
   > Per utilizzare le stesse impostazioni utilizzate in un altro gruppo di distribuzione (tra cui il nome del gruppo di distribuzione e la configurazione della distribuzione), scegliere tali impostazioni in questa pagina. Sebbene questo nuovo gruppo di distribuzione e il gruppo di distribuzione esistente possano avere lo stesso nome, li CodeDeploy considera come gruppi di distribuzione separati, poiché ciascuno è associato a un'applicazione separata.

9. In Ruolo di servizio, scegli un ruolo di servizio a cui CodeDeploy concedere l'accesso. AWS Lambda Per ulteriori informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.
10. Se si desidera utilizzare una configurazione della distribuzione predefinita, sceglierla da Deployment configuration (Configurazione di distribuzione) e passare alla fase 12. Per creare una configurazione personalizzata, proseguire con la fase successiva.

    Per ulteriori dettagli sulle configurazioni della distribuzione, consulta Configurazioni di distribuzione su un AWS Lambda piattaforma di calcolo.
11. Per creare una configurazione personalizzata, scegliere Create deployment configuration (Crea configurazione della distribuzione) ed eseguire le operazioni descritte di seguito:

    a. In Deployment configuration name (Nome configurazione della distribuzione), inserire un nome per la configurazione.

b. In Type (Tipo), scegliere un tipo di configurazione. Scegliendo Canary, il traffico viene reinstradato in due blocchi incrementali. Scegliendo Linear (Lineare), il traffico viene reinstradato in blocchi incrementali di dimensioni uguali, con un intervallo di un numero di minuti costante tra ciascun incremento.

c. In Step (Fase), inserire una percentuale di traffico, compresa tra 1 e 99, che deve essere reinstradata. Se il tipo di configurazione è Canary, questa è la percentuale di traffico che viene reinstradata con il primo blocco incrementale. Il traffico rimanente viene reinstradato con il secondo blocco incrementale dopo l'intervallo di tempo selezionato. Se il tipo di configurazione è Linear (Lineare), questa è la percentuale di traffico che viene reinstradata all'inizio di ogni intervallo.

d. In Interval (Intervallo), specificare il numero di minuti. Se il tipo di configurazione è Canary, questo è il numero di minuti tra il primo e il secondo blocco incrementale di traffico reinstradato. Se il tipo di configurazione è Linear (Lineare), questo è il numero di minuti tra ogni reinstradamento incrementale.

> **Note**
>
> La durata massima di un' AWS Lambda implementazione è di due giorni o 2.880 minuti. Pertanto, il valore massimo che è possibile specificare per il parametro Interval (Intervallo) per una configurazione di tipo canary è 2.800 minuti. Il valore massimo per una configurazione di tipo lineare dipende dal valore del parametro Step (Fase). Ad esempio, se la percentuale lineare di reinstradamento del traffico è 25%, allora sono previsti quattro blocchi incrementali di reinstradamento del traffico. Il valore massimo per il parametro Intervallo è 2.880 diviso quattro, ossia 720 minuti.

e. Selezionare Create deployment configuration (Crea configurazione della distribuzione).

12. (Facoltativo) In Advanced, configura tutte le opzioni che desideri includere nella distribuzione, come i trigger di notifica di Amazon SNS, gli CloudWatch allarmi Amazon o i rollback automatici.

    Per ulteriori informazioni, consulta Configurazione delle opzioni avanzate per un gruppo di distribuzione.

13. Scegliere Create deployment group (Crea gruppo di distribuzione).

Creare un'applicazione (CLI)

Per utilizzare il AWS CLI per creare un'applicazione, chiamate il comando create-application, specificando un nome che rappresenti in modo univoco l'applicazione. (In un AWS account, il nome di un' CodeDeploy applicazione può essere utilizzato solo una volta per regione. È possibile riutilizzare il nome di un'applicazione in diverse regioni.)

Dopo aver utilizzato il AWS CLI per creare un'applicazione, il passaggio successivo consiste nel creare un gruppo di distribuzione che specifichi le istanze in cui distribuire le revisioni. Per istruzioni, consulta Crea un gruppo di distribuzione con CodeDeploy.

Dopo aver creato il gruppo di distribuzione, il passo successivo è preparare la distribuzione di una versione rispetto all'applicazione e al gruppo di distribuzione. Per istruzioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.

Visualizza i dettagli dell'applicazione con CodeDeploy

Puoi utilizzare la CodeDeploy console AWS CLI, le o le CodeDeploy API per visualizzare i dettagli su tutte le applicazioni associate al tuo AWS account.

Argomenti

- Visualizza i dettagli dell'applicazione (console)
- Visualizza i dettagli dell'applicazione (CLI)

Visualizza i dettagli dell'applicazione (console)

Per utilizzare la CodeDeploy console per visualizzare i dettagli dell'applicazione:

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy e scegli Getting started.
3. Per visualizzare ulteriori dettagli dell'applicazione, scegliere il nome dell'applicazione nell'elenco.

Visualizza i dettagli dell'applicazione (CLI)

Per utilizzare AWS CLI per visualizzare i dettagli dell'applicazione, chiamate il get-application comando, il batch-get-application comando o il list-applications comando.

Per visualizzare i dettagli su una singola applicazione, chiamate il comando get-application, specificando il nome dell'applicazione.

Per visualizzare i dettagli su più applicazioni, chiamate il comando batch-get-applications, specificando i nomi di più applicazioni.

Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.

Creazione di una regola di notifica

È possibile utilizzare le regole di notifica per informare gli utenti in caso di modifiche apportate alle applicazioni di distribuzione, ad esempio operazioni di distribuzione riuscite e non riuscite. Le regole di notifica specificano sia gli eventi che l'argomento Amazon SNS utilizzato per inviare le notifiche. Per ulteriori informazioni, vedere Cosa sono le notifiche?

Puoi utilizzare la console o AWS CLI creare regole di notifica per AWS CodeDeploy.

Per creare una regola di notifica (console)

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy/.
2. Scegliere Application (Applicazione), quindi scegliere un'applicazione in cui si desidera aggiungere le notifiche.
3. Nella pagina dell'applicazione, scegliere Notify (Notifica), quindi selezionare Create notification rule (Crea regola di notifica). È anche possibile andare alla pagina Settings (Impostazioni) per l'applicazione e scegliere Create notification rule (Crea regola di notifica).
4. In Notification name (Nome notifica), immettere un nome per la regola.
5. In Tipo di dettaglio, scegli Basic se desideri che nella notifica siano EventBridge incluse solo le informazioni fornite ad Amazon. Scegli Completo se desideri includere le informazioni fornite ad Amazon EventBridge e le informazioni che potrebbero essere fornite da CodeDeploy o dal gestore delle notifiche.

   Per ulteriori informazioni, consulta Comprendere il contenuto e la sicurezza delle notifiche.

6. In Events that trigger notifications (Eventi che attivano le notifiche), selezionare gli eventi per i quali si desidera inviare notifiche.

Categoria	Eventi
Implementazione	Non riuscito
	Riuscito
	Avviato

7. In Targets (Destinazioni), scegli Create SNS topic (Crea argomento SNS).

> **Note**
>
> Quando crei l'argomento, viene applicata automaticamente la politica che consente CodeDeploy di pubblicare eventi sull'argomento. L'utilizzo di un argomento creato appositamente per CodeDeploy le notifiche consente inoltre di aggiungere solo gli utenti all'elenco di sottoscrizioni relativo all'argomento in cui si desidera visualizzare le notifiche relative a questa applicazione di distribuzione.

Dopo il prefisso codestar-notifications- immetti un nome per l'argomento e quindi scegli Submit (Invia).

> **Note**
>
> Se desideri utilizzare un argomento Amazon SNS esistente invece di crearne uno nuovo, in Targets (Destinazioni), scegli il relativo ARN. Assicurarsi che l'argomento disponga della policy di accesso appropriata e che l'elenco di sottoscrizioni contenga solo gli utenti autorizzati a visualizzare le informazioni sull'applicazione di distribuzione. Per ulteriori informazioni, consulta Configurare gli argomenti esistenti di Amazon SNS per le notifiche e Comprendere i contenuti e la sicurezza delle notifiche.

8. Per completare la creazione della regola, scegliere Invia.
9. È necessario iscrivere gli utenti all'argomento Amazon SNS relativo alla regola prima che possano ricevere notifiche. Per ulteriori informazioni, consulta gli argomenti relativi alla sottoscrizione degli utenti agli obiettivi di Amazon SNS. Puoi anche configurare l'integrazione tra

le notifiche e Amazon Q Developer nelle applicazioni di chat per inviare notifiche alle chat room di Amazon Chime o ai canali Slack. Per ulteriori informazioni, consulta Configurare l'integrazione tra le notifiche e Amazon Q Developer nelle applicazioni di chat.

Per creare una regola di notifica (AWS CLI)

1. Da un terminale o dal prompt dei comandi, eseguire il comando create-notification rule per generare lo skeleton JSON:

```
aws codestar-notifications create-notification-rule --generate-cli-skeleton
 > rule.json
```

   È possibile assegnare al file un nome qualsiasi. In questo esempio, il file è denominato *rule.json*.

2. Apri il file JSON in un editor di testo semplice e modificalo per includere la risorsa, i tipi di eventi e il target Amazon SNS che desideri per la regola. L'esempio seguente mostra una regola di notifica denominata **MyNotificationRule** per un'applicazione denominata *MyDeploymentApplication* in un AWS account con l'ID. *123456789012* Le notifiche vengono inviate con il tipo di dettaglio completo a un argomento di Amazon SNS denominato *codestar-notifications-MyNotificationTopic* quando le distribuzioni hanno esito positivo:

```
{
    "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"
}
```

Salvare il file.

3. Utilizzando il file appena modificato, dal terminale o dalla riga di comando, eseguire nuovamente il comando create-notification-rule per creare la regola di notifica:

```
aws codestar-notifications create-notification-rule --cli-input-json
 file://rule.json
```

4. In caso di esito positivo, il comando restituisce l'ARN della regola di notifica, simile al seguente:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Rinominare un'applicazione CodeDeploy

Puoi usare le AWS CLI o le CodeDeploy API per modificare il nome di un'applicazione.

Per visualizzare un elenco di nomi di applicazioni, utilizzare il comando AWS CLI to call il comando list-applications.

Per informazioni sull'utilizzo di per AWS CLI modificare il nome di un'applicazione, vedere update-application.

Per informazioni sull'utilizzo delle CodeDeploy API per modificare il nome di un'applicazione, consulta API_. UpdateApplication

Eliminare un'applicazione in CodeDeploy

Puoi utilizzare la CodeDeploy console AWS CLI, o un'azione CodeDeploy API per eliminare le applicazioni. Per informazioni sull'utilizzo dell'azione CodeDeploy API, consulta DeleteApplication.

> **Warning**
>
> L'eliminazione di un'applicazione rimuove le informazioni sull'applicazione dal sistema CodeDeploy, incluse tutte le informazioni del gruppo di distribuzione correlato e i dettagli della distribuzione. L'eliminazione di un'applicazione creata per una EC2/On-Premises distribuzione non rimuove alcuna revisione dell'applicazione dalle istanze né elimina le

revisioni dai bucket Amazon S3. L'eliminazione di un'applicazione creata per una EC2/On-Premises distribuzione non comporta l'interruzione di alcuna istanza Amazon EC2 né l'annullamento della registrazione delle istanze locali. Questa operazione non può essere annullata.

Argomenti

- Eliminare un'applicazione (console)
- Eliminare un'applicazione (AWS CLI)

Eliminare un'applicazione (console)

Per utilizzare la CodeDeploy console per eliminare un'applicazione:

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Nell'elenco delle applicazioni, scegli l'applicazione che desideri eliminare.

   Viene visualizzata una pagina contenente i dettagli sull'applicazione.
4. Scegli Elimina applicazione in alto a destra.
5. Quando richiesto, inserisci **`delete`** per confermare che desideri eliminare l'applicazione, quindi scegli Elimina.

Eliminare un'applicazione (AWS CLI)

Per utilizzare il AWS CLI per eliminare un'applicazione, chiamate il comando delete-application, specificando il nome dell'applicazione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.

Lavorare con i gruppi di distribuzione in CodeDeploy

È possibile specificare uno o più gruppi di distribuzione per un' CodeDeploy applicazione. Ogni distribuzione dell'applicazione utilizza uno dei suoi gruppi di distribuzione. Il gruppo di distribuzione contiene le impostazioni e le configurazioni utilizzate durante la distribuzione. La maggior parte delle impostazioni del gruppo di distribuzione dipende dalla piattaforma di elaborazione utilizzata dall'applicazione. Alcune impostazioni, come rollback, trigger e allarmi, possono essere configurate per i gruppi di distribuzione per qualsiasi piattaforma di elaborazione.

Gruppi di istanze implementate nelle implementazioni della piattaforma di calcolo Amazon ECS

In una distribuzione Amazon ECS, un gruppo di distribuzione specifica il servizio Amazon ECS, il load balancer, il listener di test opzionale e due gruppi target. Inoltre, specifica quando reindirizzare il traffico verso il set di attività sostitutivo e quando terminare il set di attività originale e l'applicazione Amazon ECS dopo una distribuzione riuscita.

Gruppi di distribuzione in AWS Lambda implementazioni di piattaforme di calcolo

In una distribuzione AWS Lambda, un gruppo di distribuzione definisce un set di CodeDeploy configurazioni per le future implementazioni di una funzione. AWS Lambda Ad esempio, il gruppo di distribuzione specifica come indirizzare il traffico verso una nuova versione di una funzione Lambda. Puoi inoltre specificare gli allarmi e i rollback. Una singola implementazione in un gruppo di distribuzione AWS Lambda può sovrascrivere una o più configurazioni di gruppo.

Gruppi di distribuzione nelle EC2/On-Premises implementazioni di Compute Platform

In una EC2/On-Premises distribuzione, un gruppo di distribuzione è un insieme di istanze individuali destinate a una distribuzione. Un gruppo di distribuzione contiene istanze con tag individuali, istanze Amazon EC2 in gruppi Amazon EC2 Auto Scaling o entrambi.

In una distribuzione in loco, le istanze nel gruppo di distribuzione sono aggiornate con l'ultima revisione dell'applicazione.

In una blue/green distribuzione, il traffico viene reindirizzato da un set di istanze a un altro annullando la registrazione delle istanze originali da uno o più sistemi di bilanciamento del carico e registrando un set sostitutivo di istanze in cui in genere è già installata la revisione più recente dell'applicazione.

È possibile associare più di un gruppo di distribuzione con un'applicazione in CodeDeploy. Questo permette di distribuire una revisione dell'applicazione a un set differente di istanze in momenti differenti. Ad esempio, è possibile utilizzare un gruppo di distribuzione per distribuire una revisione dell'applicazione a un set di istanze mirato `Test` quando si deve assicurare la qualità del codice. Successivamente, la stessa revisione dell'applicazione viene distribuita a un gruppo di distribuzione con istanze mirate `Staging` per un'ulteriore verifica. Infine, quando è pronto il rilascio dell'applicazione più recente ai clienti, viene distribuita a un gruppo di distribuzione che include istanze mirate `Production`.

È possibile anche utilizzare più gruppi di tag per perfezionare i criteri per le istanze incluse in un gruppo di distribuzione. Per informazioni, consulta Tagging Instances for Deployments.

Quando si utilizza la CodeDeploy console per creare un'applicazione, si configura contemporaneamente il primo gruppo di distribuzione. Quando si utilizza il AWS CLI per creare un'applicazione, si crea il primo gruppo di distribuzione in un passaggio separato.

Per visualizzare un elenco dei gruppi di distribuzione già associati al tuo AWS account, consultaVisualizza i dettagli del gruppo di distribuzione con CodeDeploy.

Per informazioni sui tag delle istanze Amazon EC2, consulta Lavorare con i tag utilizzando la console. Per informazioni sulle istanze locali, consulta Working with On-Premises Instances. Per informazioni su Amazon EC2 Auto Integrazione CodeDeploy con Amazon EC2 Auto Scaling Scaling, consulta.

Argomenti

- the section called “Creare un gruppo di distribuzione”
- the section called “Visualizza i dettagli del gruppo di distribuzione”
- the section called “Modificare le impostazioni del gruppo di distribuzione”
- the section called “Configurazione delle opzioni avanzate per un gruppo di distribuzione”
- the section called “Eliminare un gruppo di distribuzione”

Crea un gruppo di distribuzione con CodeDeploy

Puoi utilizzare la CodeDeploy console AWS CLI, le CodeDeploy API o un CloudFormation modello per creare gruppi di distribuzione. Per informazioni sull'utilizzo di un CloudFormation modello per creare un gruppo di distribuzione, consultaCloudFormation modelli di CodeDeploy riferimento.

Quando si utilizza la CodeDeploy console per creare un'applicazione, si configura contemporaneamente il primo gruppo di distribuzione. Quando si utilizza il AWS CLI per creare un'applicazione, si crea il primo gruppo di distribuzione in un passaggio separato.

Come parte della creazione di un gruppo di distribuzione, occorre specificare un ruolo del servizio. Per ulteriori informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.

Argomenti

- Crea un gruppo di distribuzione per una distribuzione sul posto (console)
- Crea un gruppo di distribuzione per una distribuzione (console) EC2/On-Premises blue/green
- Crea un gruppo di distribuzione per una distribuzione Amazon ECS (console)
- Configura un sistema di bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2
- Configura un sistema di bilanciamento del carico, gruppi target e listener per le distribuzioni di CodeDeploy Amazon ECS
- Creare un gruppo di distribuzione (CLI)

Crea un gruppo di distribuzione per una distribuzione sul posto (console)

Per utilizzare la CodeDeploy console per creare un gruppo di distribuzione per una distribuzione locale:

> **Warning**
>
> Non seguire i passaggi seguenti se:
>
> - Non hai preparato le istanze da utilizzare nella prima distribuzione CodeDeploy di un'applicazione. Per configurare le istanze, segui le istruzioni che trovi su Utilizzo delle istanze per CodeDeploy e quindi segui i passaggi descritti in questa sezione.
> - Desideri creare un gruppo di distribuzione che utilizza una configurazione della distribuzione personalizzata, ma non hai ancora creato la configurazione della

distribuzione. Segui le istruzioni che trovi su Create a Deployment Configuration e quindi segui i passaggi descritti in questa sezione.

- Non disponete di un ruolo di servizio che corrisponda CodeDeploy almeno alla fiducia e alle autorizzazioni descritte in. Fase 2: Creare un ruolo di servizio per CodeDeploy Per creare e configurare un ruolo di servizio, segui le istruzioni contenute in Fase 2: Creare un ruolo di servizio per CodeDeploy e quindi segui la procedura descritta in questa sezione.
- Vuoi selezionare un Classic Load Balancer, Application Load Balancer o Network Load Balancer in Elastic Load Balancing per la distribuzione sul posto, ma non l'hai ancora creato.

1. Accedi a Console di gestione AWS e apri la console all'indirizzo. CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Nella pagina Applications (Applicazioni), scegli il nome dell'applicazione per la quale desideri creare un gruppo di distribuzione.
4. Nella pagina delle applicazioni, nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione).
5. In Deployment group name (Nome gruppo di distribuzione), inserire un nome che descriva il gruppo di distribuzione.

   > **Note**
   >
   > Se desideri utilizzare le stesse impostazioni utilizzate in un altro gruppo di distribuzione (incluso il nome del gruppo di distribuzione, i tag, i nomi dei gruppi di Amazon EC2 Auto Scaling o entrambi e la configurazione di distribuzione), specifica tali impostazioni in questa pagina. Sebbene questo nuovo gruppo di distribuzione e il gruppo di distribuzione esistente abbiano lo stesso nome, li CodeDeploy tratta come gruppi di distribuzione separati, poiché ciascuno di essi è associato ad applicazioni separate.

6. In Service role (Ruolo di servizio), scegliere un ruolo di servizio che conceda a CodeDeploy l'accesso all'istanza di destinazione.
7. In Tipo di distribuzione, scegli In-place.
8. In Configurazione dell'ambiente, effettuate le seguenti operazioni:

   a. Se desideri distribuire la tua applicazione in un gruppo Amazon EC2 Auto Scaling, seleziona i gruppi Amazon EC2 Auto Scaling, quindi scegli il nome di un gruppo Amazon EC2 Auto Scaling su cui distribuire la revisione dell'applicazione. Quando vengono lanciate nuove istanze Amazon EC2 come parte di un CodeDeploy gruppo Amazon EC2 Auto Scaling, puoi distribuire automaticamente le tue revisioni alle nuove istanze. Puoi aggiungere fino a 10 gruppi Amazon EC2 Auto Scaling a un gruppo di distribuzione. Per ulteriori informazioni, consulta Integrazione CodeDeploy con Amazon EC2 Auto Scaling.

   b. Se hai selezionato i gruppi Amazon EC2 Auto Scaling, opzionalmente seleziona Aggiungi un hook di terminazione ai gruppi di Auto Scaling per installare un hook di terminazione nel tuo CodeDeploy gruppo Auto Scaling quando crei o aggiorni il gruppo di distribuzione. Una volta installato questo hook, eseguirà le distribuzioni di terminazione. CodeDeploy Per ulteriori informazioni, consulta Abilitazione delle implementazioni di terminazione durante gli eventi scale-in di Auto Scaling.

   c. Se desideri etichettare le tue istanze, seleziona le istanze o le On-premises istanze di Amazon EC2. Nei campi Chiave e Valore, inserisci i valori della coppia chiave-valore che hai usato per etichettare le istanze. A un singolo gruppo di tag puoi aggiungere fino a 10 coppie chiave-valore associate ai tag.

      i. Puoi utilizzare i caratteri jolly nel campo Valore per identificare tutte le istanze etichettate secondo determinati schemi, ad esempio nomi di istanze Amazon EC2, centri di costo e gruppi simili e così via. Ad esempio, se scegli Nome nel campo Chiave e inserisci **`GRP-*a`** nel campo Valore, CodeDeploy identifica tutte le istanze che corrispondono a quel modello, ad esempio, e. **`GRP-1a GRP-2a GRP-XYZ-a`**

      ii. Il campo Value (Valore) prevede la distinzione tra lettere maiuscole e minuscole.

      iii. Per rimuovere una coppia chiave-valore dall'elenco, scegliere l'icona di rimozione.

      Quando CodeDeploy trova le istanze che corrispondono a ciascuna coppia chiave-valore specificata o al nome del gruppo Amazon EC2 Auto Scaling, visualizza il numero di istanze corrispondenti. Per visualizzare ulteriori informazioni sulle istanze, fare clic sul numero.

Se desideri rifinire i criteri per le istanze distribuite, scegli Aggiungi gruppo di tag per creare un gruppo di tag. È possibile creare fino a tre gruppi di tag contenenti ciascuno fino a 10 coppie chiave-valore. Quando si utilizzano più gruppi di tag in un gruppo di distribuzione, sono incluse nel gruppo di distribuzione solo le istanze che sono identificate da tutti i gruppi di tag. Ciò significa che un'istanza, per essere inclusa nel gruppo di distribuzione, deve essere associata ad almeno uno dei tag in ciascuno dei gruppi.

Per informazioni su come usare i gruppi di tag per raffinare il gruppo di distribuzione, consulta Tagging Instances for Deployments.

9. Nella configurazione dell'agente con Systems Manager, specifica come desideri installare e aggiornare l' CodeDeploy agente sulle istanze del tuo gruppo di distribuzione. Per ulteriori informazioni sull' CodeDeploy agente, vedere Lavorare con l' CodeDeploy agente. Per ulteriori informazioni su Systems Manager, vedere Cos'è Systems Manager?

   a. Mai: salta la configurazione dell' CodeDeploy installazione con Systems Manager. Perché possa essere usato nelle distribuzioni, l'agente deve essere installato sulle istanze, quindi scegli questa opzione solo se desideri installare l'agente CodeDeploy in un altro modo.

   b. Solo una volta: Systems Manager installerà l' CodeDeploy agente una volta su ogni istanza del gruppo di distribuzione.

   c. Ora pianifica gli aggiornamenti: Systems Manager creerà un'associazione con State Manager che installa l' CodeDeploy agente secondo la pianificazione configurata. Per ulteriori informazioni su State Manager e associazioni, consulta Informazioni su State Manager.

10. In Deployment configuration (Configurazione della distribuzione), scegli una configurazione della distribuzione per controllare la frequenza di distribuzione delle istanze, ad esempio una per volta o tutte insieme. Per ulteriori dettagli sulle configurazioni della distribuzione, consulta Utilizzo delle configurazioni di distribuzione in CodeDeploy.

11. (Facoltativo) In Load balancer, seleziona Enable load balancing, quindi dagli elenchi seleziona i gruppi target Classic Load Balancer, Application Load Balancer e Network Load Balancer per gestire il traffico verso le istanze durante la distribuzione. CodeDeploy È possibile selezionare fino a 10 Classic Load Balancer e 10 gruppi target, per un totale di 20 articoli. Assicurati che le istanze Amazon EC2 su cui desideri effettuare la distribuzione siano registrate con i sistemi di bilanciamento del carico (Classic Load Balancer) o i gruppi target (Application Load Balancer e Network Load Balancer) selezionati.

Durante una distribuzione, le istanze originali vengono cancellate dai sistemi di bilanciamento del carico e dai gruppi target selezionati per evitare che il traffico venga indirizzato verso queste istanze durante la distribuzione. Una volta completata la distribuzione, ogni istanza viene registrata nuovamente con tutti i Classic Load Balancer e i gruppi target selezionati.

Per ulteriori informazioni sui sistemi di bilanciamento del carico per le distribuzioni, consulta. CodeDeploy Integrating CodeDeploy with Elastic Load Balancing

> ⚠ Warning
>
> Se stai configurando sia i gruppi Auto Scaling che i sistemi di bilanciamento del carico Elastic Load Balancing in questo gruppo di distribuzione e desideri collegare i sistemi di bilanciamento del carico ai gruppi Auto Scaling, ti consigliamo di completare questo allegato prima di creare la distribuzione da questo gruppo di distribuzione. CodeDeploy Il tentativo di completare l'allegato dopo aver creato la distribuzione può causare l'annullamento imprevisto della registrazione di tutte le istanze dai sistemi di bilanciamento del carico.

12. (Facoltativo) Espandi Advanced e configura tutte le opzioni che desideri includere nella distribuzione, come i trigger di notifica di Amazon SNS, gli CloudWatch allarmi Amazon, le opzioni di Auto Scaling o i rollback automatici.

    Per ulteriori informazioni, consulta Configurazione delle opzioni avanzate per un gruppo di distribuzione.

13. Scegliere Create deployment group (Crea gruppo di distribuzione).

Crea un gruppo di distribuzione per una distribuzione (console) EC2/On-Premises blue/green

Per utilizzare la CodeDeploy console per creare un gruppo di distribuzione per una blue/green distribuzione:

> ⚠ Warning
>
> Non seguire i passaggi seguenti se:

- Non sono presenti istanze con l' CodeDeploy agente installato da sostituire durante il processo di blue/green distribuzione. Per configurare le istanze, segui le istruzioni che trovi su Utilizzo delle istanze per CodeDeploy e quindi segui i passaggi descritti in questa sezione.
- Desideri creare un'applicazione che utilizza una configurazione della distribuzione personalizzata, ma non hai ancora creato la configurazione della distribuzione. Segui le istruzioni che trovi su Create a Deployment Configuration e quindi segui i passaggi descritti in questa sezione.
- Non disponete di un ruolo di servizio che corrisponda CodeDeploy almeno alla fiducia e alle autorizzazioni descritte in. Fase 2: Creare un ruolo di servizio per CodeDeploy Per creare e configurare un ruolo di servizio, segui le istruzioni contenute in Fase 2: Creare un ruolo di servizio per CodeDeploy e quindi segui la procedura descritta in questa sezione.
- Non hai creato un Classic Load Balancer o un Application Load Balancer in Elastic Load Balancing per la registrazione delle istanze nell'ambiente sostitutivo. Per ulteriori informazioni, consulta Configura un sistema di bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2.

1. Accedi a Console di gestione AWS e apri la console all'indirizzo. CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Nella pagina Applications (Applicazioni), scegli il nome dell'applicazione per la quale desideri creare un gruppo di distribuzione.
4. Nella pagina delle applicazioni, nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione).
5. In Deployment group name (Nome gruppo di distribuzione), inserire un nome che descriva il gruppo di distribuzione.

> **Note**
>
> Se desideri utilizzare le stesse impostazioni utilizzate in un altro gruppo di distribuzione (incluso il nome del gruppo di distribuzione, i tag, i nomi dei gruppi Amazon EC2 Auto Scaling e la configurazione di distribuzione), scegli tali impostazioni in questa pagina. Anche se questo nuovo gruppo di distribuzione e il gruppo di distribuzione esistente possiedono lo stesso nome, CodeDeploy li tratta come gruppi di distribuzione separati, in quanto sono associati ad applicazioni distinte.

6. In Service role (Ruolo di servizio), scegliere un ruolo di servizio che conceda a CodeDeploy l'accesso all'istanza di destinazione.
7. In Tipo di distribuzione scegli. Blue/green
8. In Configurazione dell'ambiente, procedi come segue:
   - Seleziona il metodo da utilizzare per fornire le istanze per l'ambiente sostitutivo. Sono disponibili le seguenti opzioni:
     - Copia automaticamente il gruppo Amazon EC2 Auto Scaling: crea un CodeDeploy gruppo Amazon EC2 Auto Scaling copiandone uno specificato.
     - Manually provision instances (Provisioning manuale delle istanze): non è necessario specificare le istanze per l'ambiente di sostituzione finché non si crea una distribuzione. È necessario creare le istanze prima di avviare la distribuzione. Al contrario, qui è necessario specificare le istanze che si desidera sostituire.
   - Se hai selezionato Copia automaticamente il gruppo Amazon EC2 Auto Scaling, opzionalmente seleziona Aggiungi un hook di terminazione ai CodeDeploy gruppi di Auto Scaling per installare un hook di terminazione nel tuo gruppo Auto Scaling quando crei o aggiorni il gruppo di distribuzione. Una volta installato questo hook, eseguirà le distribuzioni di terminazione. CodeDeploy Per ulteriori informazioni, consulta Abilitazione delle implementazioni di terminazione durante gli eventi scale-in di Auto Scaling.
9. Nella configurazione dell'agente con Systems Manager, specifica come desideri installare e aggiornare l' CodeDeploy agente sulle istanze del tuo gruppo di distribuzione. Per ulteriori informazioni sull' CodeDeploy agente, vedere Lavorare con l' CodeDeploy agente. Per ulteriori informazioni su Systems Manager, vedere Cos'è Systems Manager?

a. Mai: salta la configurazione dell' CodeDeploy installazione con Systems Manager. Perché possa essere usato nelle distribuzioni, l'agente deve essere installato sulle istanze, quindi scegli questa opzione solo se desideri installare l'agente CodeDeploy in un altro modo.

b. Solo una volta: Systems Manager installerà l' CodeDeploy agente una volta su ogni istanza del gruppo di distribuzione.

c. Ora pianifica gli aggiornamenti: Systems Manager creerà un'associazione con State Manager che installa l' CodeDeploy agente secondo la pianificazione configurata. Per ulteriori informazioni su State Manager e associazioni, consulta Informazioni su State Manager.

10. A seconda dell'opzione scelta nella fase 8, eseguire una delle seguenti operazioni:

    - Se hai scelto Copia automaticamente il gruppo Amazon EC2 Auto Scaling: Nel gruppo Amazon EC2 Auto Scaling, scegli o inserisci il nome del gruppo Amazon EC2 Auto Scaling che desideri utilizzare come modello per il gruppo Amazon EC2 Auto Scaling creato per le istanze nel tuo ambiente sostitutivo. Il numero di istanze attualmente integre nel gruppo Amazon EC2 Auto Scaling selezionato viene creato nell'ambiente sostitutivo.
    - Se hai scelto il provisioning manuale delle istanze: seleziona i gruppi Amazon EC2 Auto Scaling, le istanze Amazon EC2 Auto Scaling o entrambi per specificare le istanze da aggiungere a questo gruppo di distribuzione. Inserisci i valori dei tag di Amazon EC2 Auto Scaling o i nomi dei gruppi di Amazon EC2 Auto Scaling per identificare le istanze nel tuo ambiente originale (ovvero le istanze che desideri sostituire o che eseguono la revisione corrente dell'applicazione).

11. In Load balancer, seleziona Enable load balancing, quindi dagli elenchi seleziona i gruppi target Classic Load Balancer, Application Load Balancer e Network Load Balancer con cui desideri registrare le istanze Amazon EC2 sostitutive. Ogni istanza sostitutiva verrà registrata con tutti i Classic Load Balancer e i gruppi target selezionati. È possibile selezionare fino a 10 Classic Load Balancer e 10 gruppi target, per un totale di 20 articoli.

    Il traffico verrà reindirizzato dalle istanze originali a quelle sostitutive in base alle impostazioni di configurazione scelte per il reindirizzamento del traffico e la distribuzione.

    Per ulteriori informazioni sui sistemi di bilanciamento del carico per le distribuzioni, consulta. CodeDeploy Integrating CodeDeploy with Elastic Load Balancing

> **Warning**
>
> Se si configurano sia i gruppi Auto Scaling che i sistemi di bilanciamento del carico Elastic Load Balancing in questo gruppo di distribuzione e si desidera collegare i sistemi di bilanciamento del carico ai gruppi di Auto Scaling, si consiglia di completare questo allegato prima di creare la distribuzione da questo gruppo di distribuzione. CodeDeploy Il tentativo di completare l'allegato dopo aver creato la distribuzione può causare l'annullamento imprevisto della registrazione di tutte le istanze dai sistemi di bilanciamento del carico.

12. In Deployment settings (Impostazioni Distribuzione), esaminare le opzioni predefinite relative al reinstradamento del traffico verso l'ambiente di sostituzione, alla configurazione della distribuzione da usare per la distribuzione e alla modalità di gestione delle istanze dell'ambiente originale dopo la distribuzione.

    Per modificare le impostazioni, andare alla fase successiva. Altrimenti, vai al passaggio 14.

13. Per modificare le impostazioni di distribuzione per la blue/green distribuzione, scegli una delle seguenti impostazioni.

Impostazione	Opzioni
Traffic rerouting (Reinstradamento del traffico)	• Reindirizza immediatamente il traffico: non appena le istanze nell'ambiente sostitutivo vengono fornite e su di esse viene installat a la revisione più recente dell'applicazione, queste vengono registrate automatic amente presso i sistemi di bilanciamento del carico e i gruppi target specificati, provocando il reindirizzamento del traffico verso di essi. Le istanze nell'ambiente originale sono quindi rimosse. • Sceglierò se reindirizzare il traffico: le istanze nell'ambiente sostitutivo non vengono registrate con i sistemi di bilanciamento del carico e i gruppi target specificati, a meno che non si reindiriz

Impostazione	Opzioni
	zino manualmente il traffico. Se il tempo di attesa indicato trascorre senza che il traffico sia stato reinstradato, lo stato della distribuzione si modifica in Arrestata.
Configurazione dell'implementazione	Scegli la velocità con cui le istanze nell'ambiente sostitutivo vengono registrate presso i sistemi di bilanciamento del carico e i gruppi target, ad esempio una alla volta o tutte contemporaneamente. **Note** Dopo che il traffico è stato instradato correttamente verso l'ambiente di sostituzione, le istanze nell'ambiente originale sono rimosse tutte contemporaneamente indipendentemente dalla configurazione della distribuzione scelta. Per ulteriori informazioni, consulta Utilizzo delle configurazioni di distribuzione in CodeDeploy.

Impostazione	Opzioni
Istanze originali	• Termina le istanze originali nel gruppo di distribuzione: dopo che il traffico è stato reindirizzato all'ambiente sostitutivo, le istanze che sono state cancellate dai sistemi di bilanciamento del carico e dai gruppi target vengono terminate dopo il periodo di attesa specificato. Tenere presente quanto segue: • Se il gruppo Auto Scaling ha la Protezion e da Eliminazione abilitata (`prevent-force-deletion oprevent-a ll-deletion`), la CodeDeploy blue/ green distribuzione disabiliterà questa protezione sul gruppo Auto Scaling originale (blu) prima di eliminarlo durante la pulizia della distribuzione. Il gruppo Auto Scaling sostitutivo (verde) manterrà le stesse impostazioni di Protezione da Eliminazione del gruppo originale. • Mantieni attive le istanze originali del gruppo di distribuzione: dopo che il traffico è stato reindirizzato all'ambiente sostituti vo, le istanze che erano state cancellate dai sistemi di bilanciamento del carico e dai gruppi target continuano a funzionare.

14. (Facoltativo) In Advanced, configura le opzioni che desideri includere nella distribuzione, come i trigger di notifica di Amazon SNS, gli CloudWatch allarmi Amazon, le opzioni di Auto Scaling o i rollback automatici.

    Per informazioni su come specificare le opzioni avanzate nei gruppi di distribuzione di gruppi, consulta Configurazione delle opzioni avanzate per un gruppo di distribuzione.

15. Scegliere Create deployment group (Crea gruppo di distribuzione).

Crea un gruppo di distribuzione per una distribuzione Amazon ECS (console)

1. Accedi Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Dalla tabella Applications (Applicazioni), scegli il nome dell'applicazione associata al gruppo di distribuzione da modificare.
4. Nella pagina delle applicazioni, da Deployment groups (Gruppi di distribuzione), scegli il nome del gruppo di distribuzione da modificare.
5. Nella pagina delle applicazioni, nella scheda Deployment groups (Gruppi di distribuzione), scegliere Create deployment group (Crea gruppo di distribuzione). Per ulteriori informazioni su ciò di cui hai bisogno per creare un gruppo di distribuzione per una distribuzione Amazon ECS, consultaPrima di iniziare una distribuzione Amazon ECS.
6. In Deployment group name (Nome gruppo di distribuzione), inserire un nome che descriva il gruppo di distribuzione.

   > **Note**
   >
   > Per utilizzare le stesse impostazioni utilizzate in un altro gruppo di distribuzione (tra cui il nome del gruppo di distribuzione e la configurazione della distribuzione), scegliere tali impostazioni in questa pagina. Sebbene questo nuovo gruppo e il gruppo esistente possano avere lo stesso nome, li CodeDeploy considera come gruppi di distribuzione separati, poiché ciascuno è associato a un'applicazione separata.

7. In Ruolo di servizio, scegli un ruolo di servizio che garantisca CodeDeploy l'accesso ad Amazon ECS. Per ulteriori informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.
8. Da Load balancer name, scegli il nome del load balancer che serve il traffico verso il tuo servizio Amazon ECS.
9. Da Production listener port, scegli la porta e il protocollo per il listener che invia il traffico di produzione al tuo servizio Amazon ECS.

10. (Facoltativo) Da Test listener port, scegli la porta e il protocollo di un listener di test che serve il traffico verso l'attività sostitutiva impostata nel tuo servizio Amazon ECS durante la distribuzione. È possibile specificare una o più funzioni Lambda nel AppSpec file che viene eseguito durante l'hook. `AfterAllowTestTraffic` Le funzioni possono eseguire test di convalida. Se un test di convalida fallisce, viene attivato un rollback della distribuzione. Se i test di convalida riescono, viene attivato l'hook successivo nel ciclo di vita della distribuzione, `BeforeAllowTraffic`. Se non viene specificata una porta per il listener di test, non succede nulla durante l'hook. `AfterAllowTestTraffic` Per ulteriori informazioni, consulta AppSpec sezione 'hook' per una distribuzione Amazon ECS.
11. Da Nome del gruppo di destinazione 1 e Nome del gruppo di destinazione 2, scegli i gruppi target utilizzati per instradare il traffico durante la distribuzione. CodeDeploy associa un gruppo target al set di attività originale del servizio Amazon ECS e l'altro al set di attività sostitutivo. Per ulteriori informazioni, consulta Target Groups for Your Application Load Balancers.
12. Scegli Reindirizzare immediatamente il traffico o Specificare quando reindirizzare il traffico per determinare quando reindirizzare il traffico verso il servizio Amazon ECS aggiornato.

    Se scegli Reroute il traffico immediatamente, la distribuzione reindirizza automaticamente il traffico dopo il provisioning del set di attività sostitutivo.

    Se scegli Specificare quando reindirizzare il traffico, scegli il numero di giorni, ore e minuti di attesa dopo il corretto provisioning del set di attività sostitutivo. Durante questo periodo di attesa, vengono eseguiti i test di convalida nelle funzioni Lambda specificate AppSpec nel file. Se il tempo di attesa scade prima che il traffico venga reindirizzato, lo stato di distribuzione cambia in. `Stopped`
13. Per la cessazione della revisione originale, scegli il numero di giorni, ore e minuti da attendere dopo una corretta distribuzione prima che l'attività originale impostata nel tuo servizio Amazon ECS venga interrotta.
14. (Facoltativo) In Advanced, configura tutte le opzioni che desideri includere nella distribuzione, come i trigger di notifica di Amazon SNS, gli CloudWatch allarmi Amazon o i rollback automatici.

    Per ulteriori informazioni, consulta Configurazione delle opzioni avanzate per un gruppo di distribuzione.

Configura un sistema di bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2

Prima di eseguire qualsiasi blue/green distribuzione o una distribuzione sul posto per la quale si desidera specificare un load balancer opzionale nel gruppo di distribuzione, è necessario aver creato almeno un Classic Load Balancer, Application Load Balancer o Network Load Balancer in Elastic Load Balancing. Per le blue/green distribuzioni, si utilizza il sistema di bilanciamento del carico per registrare le istanze che costituiscono l'ambiente sostitutivo. Opzionalmente, le istanze nell'ambiente originale possono essere registrate con questo stesso sistema di bilanciamento del carico. Per le distribuzioni sul posto, il load balancer viene utilizzato per annullare la registrazione delle istanze su cui sta lavorando e registrarle nuovamente quando il lavoro è completo. CodeDeploy

CodeDeploy supporta blue/green e distribuisce in loco le istanze Amazon EC2 con diversi sistemi di bilanciamento del carico. Ad esempio, supponiamo di avere 200 istanze Amazon EC2, di cui 100 registrate con 2 Classic Load Balancer e altre 100 registrate con 4 gruppi target in 2 Application Load Balancer. In questo scenario, ti CodeDeploy consentirà di eseguire blue/green distribuzioni in loco su tutte le 200 istanze, anche se sono distribuite su 2 Classic Load Balancer, 2 Application Load Balancer e 4 gruppi target.

CodeDeploy supporta fino a 10 Classic Load Balancer e 10 gruppi target, per un totale di 20 articoli.

Per configurare uno o più Classic Load Balancer, segui le istruzioni riportate nel Tutorial: Create a Classic Load Balancer in User Guide for Classic Load Balancers. Tenere presente quanto segue:

- In Fase 2: definire sistema di bilanciamento del carico, nella sezione Create LB Inside (Crea sistema di bilanciamento del carico all'interno), scegli lo stesso VPC selezionato al momento della creazione delle istanze.
- Nel passaggio 5: registra le istanze EC2 con il tuo Load Balancer, seleziona le istanze attualmente presenti nel tuo gruppo di distribuzione (implementazioni sul posto) o che hai designato per l'ambiente originale (implementazioni). blue/green
- In Fase 7: creare e verificare il sistema di bilanciamento del carico, prendi nota dell'indirizzo DNS del sistema di bilanciamento del carico.

  Ad esempio, se il sistema di bilanciamento del carico è stato denominato `my-load-balancer`, il formato dell'indirizzo DNS sarà `my-load-balancer-1234567890.us-east-2.elb.amazonaws.com`.

Per configurare uno o più Application Load Balancer, segui le istruzioni in uno dei seguenti argomenti:

- Creare un Application Load Balancer
- Tutorial: Creare un Application Load Balancer utilizzando AWS CLI

Per configurare uno o più Network Load Balancer, segui le istruzioni contenute in uno dei seguenti argomenti:

- Creare un Network Load Balancer
- Tutorial: Creare un Network Load Balancer utilizzando AWS CLI

Configura un sistema di bilanciamento del carico, gruppi target e listener per le distribuzioni di CodeDeploy Amazon ECS

Prima di eseguire una distribuzione utilizzando la piattaforma di calcolo Amazon ECS, devi creare un Application Load Balancer o un Network Load Balancer, due gruppi target e uno o due listener. Questo argomento mostra come creare un Application Load Balancer. Per ulteriori informazioni, consulta Prima di iniziare una distribuzione Amazon ECS.

Uno dei gruppi target indirizza il traffico verso il set di attività originale dell'applicazione Amazon ECS. L'altro gruppo target indirizza il traffico al relativo set di attività di sostituzione. Durante la distribuzione, CodeDeploy crea un set di attività sostitutivo e reindirizza il traffico dal set di attività originale a quello nuovo. CodeDeploy determina quale gruppo target viene utilizzato per ogni set di attività.

Un listener viene utilizzato dal sistema di bilanciamento del carico per indirizzare il traffico verso i tuoi gruppi target. Un listener di produzione è obbligatorio. Puoi specificare un listener di test opzionale che indirizza il traffico verso il set di attività di sostituzione mentre esegui test di convalida.

Il sistema di bilanciamento del carico deve utilizzare un VPC con due sottoreti pubbliche in zone di disponibilità diverse. I passaggi seguenti mostrano come confermare il tuo VPC predefinito, creare un Application Load Balancer di Amazon EC2 e quindi creare due gruppi target per il tuo sistema di bilanciamento del carico. Per ulteriori informazioni, consulta Target groups for your network load balancer.

Verifica il tuo VPC, le sottoreti pubbliche e il gruppo di sicurezza predefiniti

Questo argomento mostra come creare un Application Load Balancer di Amazon EC2, due gruppi di destinazione e due porte che possono essere utilizzate durante una distribuzione di Amazon ECS.

Una delle porte è facoltativa e necessaria solo se si indirizza il traffico a una porta di test per i test di convalida durante la distribuzione.

1. Accedi a Console di gestione AWS e apri la console Amazon VPC all'indirizzo. https://console.aws.amazon.com/vpc/
2. Verifica il VPC predefinito da utilizzare. Nel pannello di navigazione, scegli Your VPCs (I tuoi VPC). Controlla quale VPC mostra Yes (Sì) nella colonna Default VPC (VPC predefinito). Questo è il VPC predefinito. Contiene le sottoreti predefinite utilizzate.
3. Scegli Subnets (Sottoreti). Annotare gli ID di sottorete di due sottoreti che mostrano Yes (Sì) nella colonna Default subnet (Sottorete predefinita). Utilizzare questi ID quando si crea il sistema di bilanciamento del carico.
4. Scegliere ogni sottorete, quindi selezionare la scheda Description (Descrizione). Verifica che le sottoreti da utilizzare siano in diverse zone di disponibilità.
5. Scegli le sottoreti, quindi scegli la scheda Route Table (Tabella di routing). Per verificare che ogni sottorete che si desidera utilizzare sia una sottorete pubblica, accertarsi che una riga con un collegamento a un Internet Gateway sia inclusa nella tabella di routing.
6. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
7. Scegli Security Groups (Gruppi di sicurezza) dal pannello di navigazione.
8. Verificare che il gruppo di sicurezza che si desidera utilizzare sia disponibile e annotare il relativo ID gruppo (ad esempio, sg-abcd1234). È possibile utilizzarlo quando si crea il sistema di bilanciamento del carico.

Crea un Application Load Balancer di Amazon EC2, due gruppi target e listener (console)

Per utilizzare la console Amazon EC2 per creare un Application Load Balancer di Amazon EC2:

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Selezionare Sistemi di bilanciamento del carico nel riquadro di navigazione.
3. Seleziona Crea sistema di bilanciamento del carico.
4. Scegli Application Load Balancer, quindi scegli Create (Crea).
5. In Name (Nome), immetti il nome del sistema di bilanciamento del carico.
6. In Scheme (Schema), scegli internet-facing.

7. In IP address type (Tipo di indirizzo IP), scegli ipv4.
8. (Facoltativo) Configurare una seconda porta listener per il sistema di bilanciamento del carico. È possibile eseguire test di convalida della distribuzione utilizzando il traffico di test che viene servito a questa porta.
    a. In Load Balancer Protocol, (Protocollo del sistema di bilanciamento del carico), scegliere Add listener (Aggiungi listener).
    b. In Load Balancer Protocol (Protocollo del sistema di bilanciamento del carico) per il secondo listener, scegli HTTP.
    c. In Load Balancer Port (Porta del sistema di bilanciamento del carico), immettere **`8080`**.
9. In Availability Zones (Zone di disponibilità), in VPC, scegliere il VPC predefinito, quindi selezionare le due sottoreti predefinite che si desidera utilizzare.
10. Seleziona Next: Configure Security Settings (Fase successiva: configurazione delle impostazioni di sicurezza).
11. Seleziona Next: Configure Security Groups (Fase successiva: configurazione dei gruppi di sicurezza).
12. Scegliere Select an existing security group (Seleziona un gruppo di sicurezza esistente), scegliere il gruppo di sicurezza predefinito, quindi annotare il relativo ID.
13. Seleziona Successivo: Configurazione del routing.
14. In Target group (Gruppo di destinazione), scegliere New target group (Nuovo gruppo di destinazione) e configurare il primo gruppo di destinazione:
    a. In Name (Nome), immetti il nome di un gruppo di destinazione (ad esempio, **`target-group-1`**).
    b. In Target type (Tipo di destinazione), scegli IP.
    c. In Protocol (Protocollo), scegli HTTP. In Port (Porta), immetti **`80`**.
    d. Seleziona Next: Register Targets (Fase successiva: registrazione delle destinazioni).
15. Scegli Next: Review (Fase successiva: revisione), quindi seleziona Create (Crea).

Per creare un secondo gruppo di destinazione per il sistema di bilanciamento del carico

1. Dopo aver effettuato il provisioning del sistema di bilanciamento del carico, apri la console Amazon EC2. Seleziona Gruppi di destinazioni nel riquadro di navigazione.
2. Scegliere Crea gruppo target.

3. In Name (Nome), immetti il nome di un gruppo di destinazione (ad esempio, **target-group-2**).
4. In Target type (Tipo di destinazione), scegli IP.
5. In Protocol (Protocollo), scegli HTTP. In Port (Porta), immetti **80**.
6. In VPC, scegli il VPC predefinito.
7. Scegli Create (Crea).

> **Note**
>
> È necessario creare due gruppi target per il sistema di bilanciamento del carico per consentire l'esecuzione della distribuzione Amazon ECS. Utilizzi l'ARN di uno dei tuoi gruppi target quando crei il tuo servizio Amazon ECS. Per ulteriori informazioni, consulta la Fase 4: Creare un servizio Amazon ECS nella Amazon ECS User Guide.

Crea un Application Load Balancer di Amazon EC2, due gruppi target e listener (CLI)

Per creare un Application Load Balancer utilizzando: AWS CLI

1. Utilizza il comando create-load-balancer per creare un Application Load Balancer. Specificare due sottoreti che non si trovano nella stessa zona di disponibilità e un gruppo di sicurezza.

```
aws elbv2 create-load-balancer --name bluegreen-alb \
--subnets subnet-abcd1234 subnet-abcd5678 --security-groups sg-abcd1234 --
region us-east-1
```

   L'output include l'Amazon Resource Name (ARN) del sistema di bilanciamento del carico, con il formato seguente:

```
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642
```

2. Utilizzare il comando create-target-group per creare il primo gruppo target. CodeDeploy instrada il traffico di questo gruppo target al set di attività originale o di sostituzione nel servizio.

```
aws elbv2 create-target-group --name bluegreentarget1 --protocol HTTP --port 80 \
--target-type ip --vpc-id vpc-abcd1234 --region us-east-1
```

   L'output include l'ARN del primo gruppo target, con il seguente formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4
```

3. Utilizzare il comando create-target-group per creare il secondo gruppo target. CodeDeploy instrada il traffico del gruppo target al set di attività non servito dal primo gruppo target. Ad esempio, se il primo gruppo target instrada il traffico al set di attività originale, questo gruppo target instrada il traffico al set di attività di sostituzione.

```
aws elbv2 create-target-group --name bluegreentarget2 --protocol HTTP --port 80 \
--target-type ip --vpc-id vpc-abcd1234 --region us-east-1
```

L'output include l'ARN del secondo gruppo target, con il seguente formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget2/209a844cd01825a4
```

4. Utilizzare il comando create-listener per creare un listener con una regola predefinita che inoltra il traffico di produzione alla porta 80.

```
aws elbv2 create-listener --load-balancer-arn
 arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642 \
--protocol HTTP --port 80 \
--default-actions
 Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup
bluegreentarget1/209a844cd01825a4 --region us-east-1
```

L'output include l'ARN del listener, con il formato seguente:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

5. (Facoltativo) Utilizzare il comando create-listener per creare un secondo listener con una regola predefinita che inoltra il traffico di test alla porta 8080. È possibile eseguire test di convalida della distribuzione utilizzando il traffico di test servito da questa porta.

```
aws elbv2 create-listener --load-balancer-arn
 arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642 \
```

```
--protocol HTTP --port 8080 \
--default-actions
 Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup
bluegreentarget2/209a844cd01825a4 --region us-east-1
```

L'output include l'ARN del listener, con il formato seguente:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

Creare un gruppo di distribuzione (CLI)

Per utilizzarlo per AWS CLI creare un gruppo di distribuzione, chiama il comando create-deployment-group, specificando:

- Nome dell'applicazione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.
- Un nome per il gruppo di distribuzione. Viene creato un gruppo di distribuzione con questo nome per l'applicazione specifica. Un gruppo di distribuzione può essere associato solamente a una sola applicazione.
- Informazioni sui tag, sui gruppi di tag o sui nomi dei gruppi Amazon EC2 Auto Scaling che identificano le istanze da includere nel gruppo di distribuzione.
- L'identificatore Amazon Resource Name (ARN) del ruolo di servizio che consente di agire CodeDeploy per conto del tuo AWS account quando interagisci con altri servizi. AWS Per ottenere l'ARN del ruolo del servizio, consulta Ottieni il ruolo di servizio ARN (CLI). Per ulteriori informazioni sui ruoli di servizio, consulta i termini e i concetti relativi ai ruoli nella IAM User Guide.
- Informazioni sul tipo di implementazione, in loco o blue/green, da associare al gruppo di distribuzione.
- (Facoltativo) Il nome di una configurazione della distribuzione esistente. Per visualizzare un elenco di configurazioni, consulta View Deployment Configuration Details. Se non specificato, CodeDeploy utilizza una configurazione della distribuzione predefinita.
- (Facoltativo) Comandi per creare un trigger che invia notifiche sugli eventi di distribuzione e istanza a coloro che sono abbonati a un argomento di Amazon Simple Notification Service. Per ulteriori informazioni, consulta Monitoring Deployments with Amazon SNS Event Notifications.

- (Facoltativo) Comandi per aggiungere CloudWatch allarmi esistenti al gruppo di distribuzione che vengono attivati se una metrica specificata in un allarme scende al di sotto o supera una soglia definita.
- (Facoltativo) Comandi per una distribuzione per ripristinare l'ultima revisione valida conosciuta quando una distribuzione fallisce o viene attivato un CloudWatch allarme.
- (Facoltativo) Comandi per una distribuzione per generare hook di eventi del ciclo di vita durante un evento scale-in di Auto Scaling. Per ulteriori informazioni, consulta Come funziona Amazon EC2 Auto Scaling con CodeDeploy.
- Per distribuzioni in loco:
  - (Facoltativo) I nomi dei sistemi Classic Load Balancer, Application Load Balancer o Network Load Balancer in Elastic Load Balancing che gestiscono il traffico verso le istanze durante i processi di distribuzione.
- Per le distribuzioni: blue/green
  - Configurazione del processo di blue/green distribuzione:
    - Come viene eseguito il provisioning di nuove istanze nell'ambiente di sostituzione.
    - Se reindirizzare il traffico verso l'ambiente di sostituzione immediatamente o attendere un periodo di tempo specificato per la reinstradazione manuale del traffico.
    - Se le istanze nell'ambiente originale devono essere terminate.
  - I nomi dei sistemi Classic Load Balancer, Application Load Balancer o Network Load Balancer in Elastic Load Balancing da utilizzare per le istanze registrate nell'ambiente sostitutivo.

> **Warning**
>
> Se stai configurando sia un gruppo Auto Scaling che un load balancer Elastic Load Balancing nel tuo gruppo di distribuzione e desideri collegare il load balancer al gruppo Auto Scaling, ti consigliamo di completare questo allegato CodeDeploy prima di creare la distribuzione da questo gruppo di distribuzione. Il tentativo di completare l'allegato dopo aver creato la distribuzione può causare l'annullamento imprevisto della registrazione di tutte le istanze dal sistema di bilanciamento del carico.

Visualizza i dettagli del gruppo di distribuzione con CodeDeploy

Puoi utilizzare la CodeDeploy console AWS CLI, le o le CodeDeploy API per visualizzare i dettagli su tutti i gruppi di distribuzione associati a un'applicazione.

Argomenti

- Visualizza i dettagli del gruppo di distribuzione (console)
- Visualizza i dettagli del gruppo di distribuzione (CLI)

Visualizza i dettagli del gruppo di distribuzione (console)

Per utilizzare la CodeDeploy console per visualizzare i dettagli del gruppo di distribuzione:

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Nella pagina Applications (Applicazioni), scegli il nome dell'applicazione associato al gruppo di distribuzione.

   > **Note**
   >
   > Se non viene visualizzata alcuna voce, assicurati che sia selezionata la regione corretta. Nella barra di navigazione, nel selettore di regione, scegli una delle regioni elencate in Regione ed endpoint in. Riferimenti generali di AWS CodeDeploy è supportato solo in queste regioni.

4. Per visualizzare i dettagli relativi a un singolo gruppo di distribuzione, nella scheda Deployment groups (Gruppi di distribuzione), scegli il nome del gruppo di distribuzione.

Visualizza i dettagli del gruppo di distribuzione (CLI)

Per utilizzare il comando AWS CLI per visualizzare i dettagli del gruppo di distribuzione, chiama il `get-deployment-group` comando o il `list-deployment-groups` comando.

Per visualizzare i dettagli su un singolo gruppo di distribuzione, chiamate il comando get-deployment-group, specificando:

- Il nome dell'applicazione associato al gruppo di distribuzione. Per ottenere il nome dell'applicazione, chiamate il comando list-applications.
- Il nome del gruppo di distribuzione. Per ottenere il nome del gruppo di distribuzione, chiamate il comando list-deployment-groups.

Per visualizzare un elenco di nomi di gruppi di distribuzione, chiamate il comando list-deployment-groups, specificando il nome dell'applicazione associato ai gruppi di distribuzione. Per ottenere il nome dell'applicazione, chiama il comando list-applications.

Modifica le impostazioni del gruppo di distribuzione con CodeDeploy

Puoi utilizzare la CodeDeploy console AWS CLI, o le CodeDeploy API per modificare le impostazioni di un gruppo di distribuzione.

> **Warning**
>
> Non seguire queste fasi se desideri che il gruppo di distribuzione utilizzi un gruppo di distribuzione personalizzato non ancora creato. Segui invece le istruzioni riportate in Create a Deployment Configuration, quindi torna alla procedura descritta in questo argomento.
> Non seguire queste fasi se desideri che il gruppo di distribuzione utilizzi un ruolo di servizio diverso non ancora creato. Il ruolo di servizio deve CodeDeploy disporre almeno delle autorizzazioni descritte in. Fase 2: Creare un ruolo di servizio per CodeDeploy Per creare e configurare un ruolo di servizio con le autorizzazioni appropriate, segui le istruzioni contenute in Fase 2: Creare un ruolo di servizio per CodeDeploy e quindi torna alla procedura descritta in questo argomento.

Argomenti

- Modifica le impostazioni del gruppo di distribuzione (console)
- Modifica delle impostazioni del gruppo di distribuzione (CLI)

Modifica le impostazioni del gruppo di distribuzione (console)

Per utilizzare la CodeDeploy console per modificare le impostazioni del gruppo di distribuzione:

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Nell'elenco delle applicazioni, scegli il nome dell'applicazione associata al gruppo di distribuzione da modificare.

   > **Note**
   >
   > Se non viene visualizzata alcuna voce, assicurati che sia selezionata la regione corretta. Nella barra di navigazione, nel selettore di regione, scegli una delle regioni elencate in Regione ed endpoint in. Riferimenti generali di AWS CodeDeploy è supportato solo in queste regioni.

4. Scegli la scheda Deployment groups (Gruppi di distribuzione), quindi scegli il nome del gruppo di distribuzione da modificare.
5. Nella pagina del gruppo di distribuzione, scegli Modifica.
6. Modifica le opzioni del gruppo di distribuzione.

   Per informazioni sui componenti del gruppo di distribuzione, consulta Crea un gruppo di distribuzione con CodeDeploy.

7. Scegli Save changes (Salva modifiche).

Modifica delle impostazioni del gruppo di distribuzione (CLI)

Per utilizzare AWS CLI per modificare le impostazioni del gruppo di distribuzione, chiamate il comando update-deployment-group, specificando:

- Per le EC2/On-Premises implementazioni e AWS Lambda:
  - Nome dell'applicazione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.

- Il nome del gruppo di distribuzione corrente. Per visualizzare un elenco di nomi di gruppi di distribuzione, chiamate il comando list-deployment-groups.
- (Facoltativo) un nome diverso del gruppo di distribuzione.
- (Facoltativo) Un diverso Amazon Resource Name (ARN) che corrisponde a un ruolo di servizio che consente di agire CodeDeploy per AWS conto del tuo account quando interagisci con altri servizi. AWS Per ottenere l'ARN del ruolo del servizio, consulta Ottieni il ruolo di servizio ARN (CLI). Per ulteriori informazioni sui ruoli di servizio, consulta i termini e i concetti relativi ai ruoli nella Guida per l'utente IAM.
- (Facoltativo) il nome di una configurazione della distribuzione. Per visualizzare un elenco di configurazioni, consulta View Deployment Configuration Details. (Se non specificato, CodeDeploy utilizza una configurazione di distribuzione predefinita.)
- (Facoltativo) Comandi per aggiungere uno o più CloudWatch allarmi esistenti al gruppo di distribuzione che vengono attivati se una metrica specificata in un allarme scende al di sotto o supera una soglia definita.
- (Facoltativo) Comandi per una distribuzione per ripristinare l'ultima revisione valida conosciuta quando una distribuzione fallisce o viene attivato un CloudWatch allarme.
- (Facoltativo) Comandi per una distribuzione per generare hook di eventi del ciclo di vita durante un evento scale-in di Auto Scaling. Per ulteriori informazioni, consulta Come funziona Amazon EC2 Auto Scaling con CodeDeploy.
- (Facoltativo) Comandi per creare o aggiornare un trigger da pubblicare su un argomento in Amazon Simple Notification Service, in modo che gli abbonati a quell'argomento ricevano notifiche sugli eventi di distribuzione e istanza in questo gruppo di distribuzione. Per informazioni, consulta Monitoring Deployments with Amazon SNS Event Notifications.

- Solo per le EC2/On-Premises distribuzioni:
  - (Facoltativo) tag o gruppi di tag di sostituzione che identificano in modo univoco le istanze da includere nel gruppo di distribuzione.
  - (Facoltativo) I nomi dei gruppi Amazon EC2 Auto Scaling sostitutivi da aggiungere al gruppo di distribuzione.
- Solo per le distribuzioni Amazon ECS:
  - Il servizio Amazon ECS da distribuire.
  - Informazioni sul sistema di bilanciamento del carico, tra cui Application Load Balancer o Network Load Balancer, i gruppi target necessari per una distribuzione di Amazon ECS e informazioni sulla produzione e sui listener di test opzionali.

Configurazione delle opzioni avanzate per un gruppo di distribuzione

Quando si crea o si aggiorna un gruppo di distribuzione, è possibile configurare una serie di opzioni per offrire maggiore controllo e ottenere una visione più precisa delle distribuzioni per il gruppo specifico.

Utilizza le informazioni contenute in questa pagina per configurare le opzioni avanzate quando lavori con i gruppi di distribuzione nei seguenti argomenti:

- Crea un'applicazione con CodeDeploy
- Crea un gruppo di distribuzione con CodeDeploy
- Modifica le impostazioni del gruppo di distribuzione con CodeDeploy

Trigger di notifica di Amazon SNS: puoi aggiungere trigger a un gruppo di CodeDeploy distribuzione per ricevere notifiche sugli eventi relativi alle distribuzioni in quel gruppo di distribuzione. Queste notifiche vengono inviate ai destinatari abbonati a un argomento di Amazon SNS che hai inserito nell'azione del trigger.

Devi aver già impostato l'argomento Amazon SNS a cui indirizzerà questo trigger e CodeDeploy devi avere l'autorizzazione a pubblicare sull'argomento da questo gruppo di distribuzione. Se non hai ancora completato queste operazioni di configurazione, puoi aggiungere trigger al gruppo di distribuzione successivamente.

Se desideri creare un trigger ora per ricevere notifiche relative agli eventi di distribuzione nel gruppo di distribuzione per questa applicazione, scegli Create trigger (Crea trigger).

Se la tua distribuzione è su un'istanza Amazon EC2, puoi creare e ricevere notifiche relative alle istanze.

Per ulteriori informazioni, consulta Monitoring Deployments with Amazon SNS Event Notifications.

CloudWatch Allarmi Amazon: puoi creare un CloudWatch allarme che monitora una singola metrica in un periodo di tempo specificato ed esegue una o più azioni in base al valore della metrica rispetto a una determinata soglia per diversi periodi di tempo. Per un'implementazione Amazon EC2, puoi creare un allarme per un'istanza o un gruppo Amazon EC2 Auto Scaling che utilizzi nelle tue operazioni. CodeDeploy Per una distribuzione AWS Lambda e Amazon ECS, puoi creare un allarme per errori in una funzione Lambda.

Puoi configurare una distribuzione in modo che si interrompa quando un CloudWatch allarme Amazon rileva che una metrica è scesa al di sotto o ha superato una soglia definita.

Devi aver già creato l'allarme CloudWatch prima di poterlo aggiungere a un gruppo di distribuzione.

1. Per aggiungere il monitoraggio dell'allarme al gruppo di distribuzione, in Alarms (Allarmi) selezionare Add alarm (Aggiungi allarme).
2. Inserisci il nome di un CloudWatch allarme che hai già configurato per monitorare questa distribuzione.

   È necessario inserire l' CloudWatch allarme esattamente come è stato creato in CloudWatch. Per visualizzare un elenco di allarmi, apri la CloudWatch console all'indirizzohttps://console.aws.amazon.com/cloudwatch/, quindi scegli ALLARME.

Opzioni aggiuntive:

- Se si desidera che le distribuzioni procedano senza tenere conto degli allarmi aggiunti, scegliere Ignore alarm configuration (Ignora configurazione allarme).

  Questa opzione è utile per disattivare temporaneamente il monitoraggio dell'allarme per un gruppo di distribuzione senza dover aggiungere gli stessi allarmi in un secondo momento.
- (Facoltativo) Se desideri che le distribuzioni continuino nel caso in cui non CodeDeploy sia possibile recuperare lo stato dell'allarme da Amazon CloudWatch, scegli Continua le distribuzioni anche se lo stato dell'allarme non è disponibile.

> **Note**
>
> Questa opzione corrisponde all'oggetto ignorePollAlarmFailure dell'AlarmConfigurationAPI. CodeDeploy

Per ulteriori informazioni, consulta Monitoraggio delle distribuzioni con allarmi in CloudWatch CodeDeploy.

Rollback automatici: è possibile configurare un gruppo di distribuzione o una distribuzione in modo che esegua automaticamente il rollback quando una distribuzione non riesce o quando viene raggiunta una soglia di monitoraggio specificata. In questo caso, viene distribuita la versione più recente di una revisione di applicazione valida nota. È possibile configurare le impostazioni facoltative

per un gruppo di distribuzione quando si utilizza la console per creare un'applicazione o per creare o aggiornare un gruppo di distribuzione. Quando si crea una nuova distribuzione, è possibile anche scegliere di ignorare la configurazione di rollback automatico specificata per il gruppo di distribuzione.

- È possibile abilitare le distribuzioni per eseguire il rollback alla revisione valida nota più recente in caso di problemi scegliendo uno o entrambi i seguenti modi:
  - Effettua il rollback quando una distribuzione fallisce. CodeDeploy ridistribuirà l'ultima revisione valida conosciuta come nuova distribuzione.
  - Roll back when alarm thresholds are met (Esegui rollback quando vengono raggiunte le soglie di allarme). Se hai aggiunto un allarme a questa applicazione nel passaggio precedente, CodeDeploy ridistribuirà l'ultima revisione valida conosciuta quando vengono attivati uno o più degli allarmi specificati.

  > **Note**
  >
  > Per ignorare temporaneamente una configurazione di rollback, scegliere Disable rollbacks (Disabilita rollback). Questa opzione è utile quando si desidera disabilitare temporaneamente i rollback automatici senza dover eseguire nuovamente la stessa configurazione in un secondo momento.

  Per ulteriori informazioni, consulta Ridistribuisci e ripristina una distribuzione con CodeDeploy.

Aggiornamenti automatici alle istanze obsolete: in determinate circostanze, CodeDeploy può distribuire una revisione obsoleta dell'applicazione sulle istanze Amazon EC2. Ad esempio, se le istanze EC2 vengono avviate in un gruppo di Auto Scaling (ASG) mentre è in corso CodeDeploy una distribuzione, tali istanze ricevono la revisione precedente dell'applicazione anziché quella più recente. Per aggiornare tali istanze, avvia CodeDeploy automaticamente una distribuzione successiva (subito dopo la prima) per aggiornare eventuali istanze obsolete. Se desideri modificare questo comportamento predefinito in modo che le istanze EC2 obsolete rimangano nella versione precedente, puoi farlo tramite l' CodeDeploy API o la (AWS Command Line Interface CLI).

Per configurare gli aggiornamenti automatici delle istanze obsolete tramite l'API, includi il parametro request nell'azione `outdatedInstancesStrategy` o. `UpdateDeploymentGroup` `CreateDeploymentGroup` Per i dettagli, consulta l'AWS CodeDeploy API Reference.

Per configurare gli aggiornamenti automatici tramite AWS CLI, utilizza uno dei seguenti comandi:

```
aws deploy update-deployment-group arguments --outdated-instances-strategy UPDATE|IGNORE
```

Oppure...

```
aws deploy create-deployment-group arguments --outdated-instances-strategy UPDATE|IGNORE
```

... dove *arguments* viene sostituito con gli argomenti richiesti per la distribuzione e *UPDATE|IGNORE* viene sostituito con per UPDATE abilitare gli aggiornamenti automatici o IGNORE per disabilitarli.

Esempio:

```
aws deploy update-deployment-group --application-name "MyApp" --current-deployment-group-name "MyDG" --region us-east-1 --outdated-instances-strategy IGNORE
```

Per i dettagli su questi AWS CLI comandi, consulta il AWS CLI Command Reference.

Eliminare un gruppo di distribuzione con CodeDeploy

Puoi utilizzare la CodeDeploy console AWS CLI, le o le CodeDeploy API per eliminare i gruppi di distribuzione associati al tuo AWS account.

> **Warning**
>
> Se elimini un gruppo di distribuzione, verranno eliminati anche tutti i dettagli associati a quel gruppo di CodeDeploy distribuzione. Le istanze utilizzate nel gruppo di distribuzione rimangono invariate. Questa operazione non può essere annullata.

Argomenti

- Eliminare un gruppo di distribuzione (console)
- Eliminare un gruppo di distribuzione (CLI)

Eliminare un gruppo di distribuzione (console)

Per utilizzare la CodeDeploy console per eliminare un gruppo di distribuzione:

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Nell'elenco delle applicazioni, scegli il nome dell'applicazione associata al gruppo di distribuzione.
4. Nella pagina Application details (Dettagli applicazione), nella scheda Deployment groups (Gruppi di distribuzione), scegli il nome del gruppo di distribuzione da eliminare.
5. Nella pagina Deployment details (Dettagli sulla distribuzione), scegli Delete (Elimina).
6. Quando richiesto, digita il nome del gruppo di distribuzione per confermare che desideri eliminarlo e quindi scegli Delete (Elimina).

Eliminare un gruppo di distribuzione (CLI)

Per utilizzare per AWS CLI eliminare un gruppo di distribuzione, chiamate il comando delete-deployment-group, specificando:

- Il nome dell'applicazione associata a questo gruppo di distribuzione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.
- Il nome del gruppo di distribuzione associato a questa applicazione. Per visualizzare un elenco di nomi di gruppi di distribuzione, chiamate il comando list-deployment-groups.

Utilizzo delle revisioni delle applicazioni per CodeDeploy

In CodeDeploy, una revisione contiene una versione dei file sorgente che CodeDeploy verrà distribuita nelle tue istanze o gli script verranno eseguiti sulle tue istanze. CodeDeploy

Pianifica la revisione, aggiungi un AppSpec file alla revisione e quindi invia la revisione ad Amazon S3 o. GitHub Dopo aver inviato la revisione, è possibile distribuirla.

Argomenti

- Pianifica una revisione per CodeDeploy
- Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy
- Scegli un tipo di CodeDeploy repository
- Invia una revisione CodeDeploy ad Amazon S3 EC2/On-Premises (solo distribuzioni)
- Visualizza i dettagli della revisione dell'applicazione con CodeDeploy
- Registra una revisione dell'applicazione in Amazon S3 con CodeDeploy

Pianifica una revisione per CodeDeploy

Una buona pianificazione semplifica molto la distribuzione di revisioni.

Per le distribuzioni su una piattaforma di calcolo AWS Lambda o Amazon ECS, una revisione è la stessa del file. AppSpec Le seguenti informazioni non si applicano. Per ulteriori informazioni, consulta Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy

Per le distribuzioni su una piattaforma di EC2/On-Premises elaborazione, inizia creando una directory principale (cartella) vuota sulla macchina di sviluppo. Questa è la cartella in cui memorizzare i file di origine (ad esempio testo e file binari, eseguibili, pacchetti e così via) per essere distribuiti alle istanze o gli script da eseguire sulle istanze.

Ad esempio, nella cartella `/tmp/` principale di Linux, macOS o Unix o nella cartella `c:\temp` principale di Windows:

```
/tmp/ or c:\temp (root folder)
  |--content (subfolder)
  |    |--myTextFile.txt
  |    |--mySourceFile.rb
  |    |--myExecutableFile.exe
  |    |--myInstallerFile.msi
```

```
|    |--myPackage.rpm
|    |--myImageFile.png
|--scripts (subfolder)
|    |--myShellScript.sh
|    |--myBatchScript.bat
|    |--myPowerShellScript.ps1
|--appspec.yml
```

La cartella principale deve includere anche un file con le specifiche dell'applicazione (AppSpec file), come illustrato di seguito. Per ulteriori informazioni, consulta Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy.

Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy

Questo argomento mostra come aggiungere un AppSpec file alla distribuzione. Include anche modelli per creare un AppSpec file per una AWS Lambda e EC2/On-Premises una distribuzione.

Argomenti

- Aggiungi un AppSpec file per una distribuzione Amazon ECS
- Aggiungi un AppSpec file per un AWS Implementazione Lambda
- Aggiungi un AppSpec file per una distribuzione EC2/On-Premises

Aggiungi un AppSpec file per una distribuzione Amazon ECS

Per una distribuzione su una piattaforma di calcolo Amazon ECS:

- Il AppSpec file specifica la definizione del task di Amazon ECS utilizzata per la distribuzione, una mappatura del nome del contenitore e delle porte utilizzata per instradare il traffico e le funzioni Lambda opzionali eseguite dopo gli eventi del ciclo di vita della distribuzione.
- Una revisione è uguale a un file. AppSpec
- Un AppSpec file può essere scritto usando JSON o YAML.
- Un AppSpec file può essere salvato come file di testo o inserito direttamente in una console quando si crea una distribuzione. Per ulteriori informazioni, consulta Crea un'implementazione di Amazon ECS Compute Platform (console).

Per creare un AppSpec file

1. Copia il modello JSON o YAML in un editor di testo o nell' AppSpec editor della console.
2. Modifica il modello in base alle esigenze.
3. Usa un validatore JSON o YAML per convalidare il file. AppSpec Se utilizzi l' AppSpeceditor, il file viene convalidato quando scegli Crea distribuzione.
4. Se si utilizza un editor di testo, salvare il file. Se utilizzi il AWS CLI per creare la tua distribuzione, fai riferimento al AppSpec file se si trova sul tuo disco rigido o in un bucket Amazon S3. Se usi la console, devi inviare il AppSpec file ad Amazon S3.

Modello di AppSpec file YAML per una distribuzione Amazon ECS con istruzioni

Di seguito è riportato un modello YAML di un AppSpec file per una distribuzione Amazon ECS con tutte le opzioni disponibili. Per informazioni sugli eventi del ciclo di vita da utilizzare nella sezione `hooks`, consulta AppSpec sezione 'hook' per una distribuzione Amazon ECS.

```
# This is an appspec.yml template file for use with an Amazon ECS deployment in
 CodeDeploy.
# The lines in this template that start with the hashtag are
#   comments that can be safely left in the file or
#   ignored.
# For help completing this file, see the "AppSpec File Reference" in the
#   "CodeDeploy User Guide" at
#   https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
# In the Resources section, you must specify the following: the Amazon ECS service,
 task definition name,
# and the name and port of the load balancer to route traffic,
# target version, and (optional) the current version of your AWS Lambda function.
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "" # Specify the ARN of your task definition
 (arn:aws:ecs:region:account-id:task-definition/task-definition-family-name:task-
definition-revision-number)
        LoadBalancerInfo:
          ContainerName: "" # Specify the name of your Amazon ECS application's
 container
```

```
          ContainerPort: "" # Specify the port for your container where traffic
 reroutes
# Optional properties
        PlatformVersion: "" # Specify the version of your Amazon ECS Service
        NetworkConfiguration:
          AwsvpcConfiguration:
            Subnets: ["",""] # Specify one or more comma-separated subnets in your
 Amazon ECS service
            SecurityGroups: ["",""] # Specify one or more comma-separated security
 groups in your Amazon ECS service
            AssignPublicIp: "" # Specify "ENABLED" or "DISABLED"
# (Optional) In the Hooks section, specify a validation Lambda function to run during
# a lifecycle event.
Hooks:
# Hooks for Amazon ECS deployments are:
    - BeforeInstall: "" # Specify a Lambda function name or ARN
    - AfterInstall: "" # Specify a Lambda function name or ARN
    - AfterAllowTestTraffic: "" # Specify a Lambda function name or ARN
    - BeforeAllowTraffic: "" # Specify a Lambda function name or ARN
    - AfterAllowTraffic: "" # Specify a Lambda function name or ARN
```

AppSpec File JSON per un modello di distribuzione Amazon ECS

Di seguito è riportato un modello JSON per un AppSpec file per una distribuzione Amazon ECS con tutte le opzioni disponibili. Per le istruzioni relative al modello, consulta i commenti nella versione YAML nella sezione precedente. Per informazioni sugli eventi del ciclo di vita da utilizzare nella sezione `hooks`, consulta AppSpec sezione 'hook' per una distribuzione Amazon ECS.

```
{
 "version": 0.0,
 "Resources": [
  {
   "TargetService": {
    "Type": "AWS::ECS::Service",
    "Properties": {
       "TaskDefinition": "",
       "LoadBalancerInfo": {
        "ContainerName": "",
        "ContainerPort":
       },
       "PlatformVersion": "",
       "NetworkConfiguration": {
        "AwsvpcConfiguration": {
```

```
          "Subnets": [
           "",
           ""
          ],
          "SecurityGroups": [
           "",
           ""
          ],
          "AssignPublicIp": ""
         }
        }
      }
    }
  }
 ],
 "Hooks": [
  {
   "BeforeInstall": ""
  },
  {
   "AfterInstall": ""
  },
  {
   "AfterAllowTestTraffic": ""
  },
  {
   "BeforeAllowTraffic": ""
  },
  {
   "AfterAllowTraffic": ""
  }
 ]
}
```

Aggiungi un AppSpec file per un AWS Implementazione Lambda

Per una distribuzione su una piattaforma di elaborazione AWS Lambda:

- Il AppSpec file contiene istruzioni sulle funzioni Lambda da distribuire e utilizzare per la convalida della distribuzione.
- Una revisione è uguale a un file. AppSpec
- Un AppSpec file può essere scritto usando JSON o YAML.

- Un AppSpec file può essere salvato come file di testo o inserito direttamente in un AppSpec editor di console durante la creazione di una distribuzione. Per ulteriori informazioni, consulta Crea un AWS Implementazione della piattaforma di elaborazione Lambda (console).

Per creare un AppSpec file:

1. Copia il modello JSON o YAML in un editor di testo o nell' AppSpec editor della console.
2. Modifica il modello in base alle esigenze.
3. Usa un validatore JSON o YAML per convalidare il file. AppSpec Se utilizzi l' AppSpeceditor, il file viene convalidato quando scegli Crea distribuzione.
4. Se si utilizza un editor di testo, salvare il file. Se utilizzi il AWS CLI per creare la tua distribuzione, fai riferimento al AppSpec file se si trova sul tuo disco rigido o in un bucket Amazon S3. Se usi la console, devi inviare il AppSpec file ad Amazon S3.

Modello di AppSpec file YAML per un AWS Lambda distribuzione con istruzioni

Per informazioni sugli eventi del ciclo di vita da utilizzare nella sezione Hooks, consulta AppSpec sezione 'hooks' per un AWS Implementazione Lambda.

```
# This is an appspec.yml template file for use with an AWS Lambda deployment in
 CodeDeploy.
# The lines in this template starting with the hashtag symbol are
#   instructional comments and can be safely left in the file or
#   ignored.
# For help completing this file, see the "AppSpec File Reference" in the
#   "CodeDeploy User Guide" at
#   https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
# In the Resources section specify the name, alias,
# target version, and (optional) the current version of your AWS Lambda function.
Resources:
  - MyFunction: # Replace "MyFunction" with the name of your Lambda function
      Type: AWS::Lambda::Function
      Properties:
        Name: "" # Specify the name of your Lambda function
        Alias: "" # Specify the alias for your Lambda function
        CurrentVersion: "" # Specify the current version of your Lambda function
        TargetVersion: "" # Specify the version of your Lambda function to deploy
# (Optional) In the Hooks section, specify a validation Lambda function to run during
```

```
# a lifecycle event. Replace "LifeCycleEvent" with BeforeAllowTraffic
# or AfterAllowTraffic.
Hooks:
    - LifeCycleEvent: "" # Specify a Lambda validation function between double-quotes.
```

AppSpec file JSON per un AWS Lambda modello di distribuzione

Nel modello seguente, sostituisci "MyFunction" con il nome della tua AWS Lambda funzione. Nella sezione Hooks opzionale, sostituisci gli eventi del ciclo di vita con o. BeforeAllowTraffic AfterAllowTraffic

Per informazioni sugli eventi del ciclo di vita da utilizzare nella sezione Hooks, consulta AppSpec sezione 'hooks' per un AWS Implementazione Lambda.

```
{
  "version": 0.0,
  "Resources": [{
   "MyFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
     "Name": "",
     "Alias": "",
     "CurrentVersion": "",
     "TargetVersion": ""
    }
   }
  }],
  "Hooks": [{
    "LifeCycleEvent": ""
   }
  ]
 }
```

Aggiungi un AppSpec file per una distribuzione EC2/On-Premises

Senza un AppSpec file, CodeDeploy non è possibile mappare i file di origine nella revisione dell'applicazione alle relative destinazioni o eseguire script per la distribuzione su una piattaforma di EC2/On-Premises elaborazione.

Ogni revisione deve contenere un solo file. AppSpec

Per aggiungere un AppSpec file a una revisione:

1. Copiare il modello in un editor di testo.
2. Modifica il modello in base alle esigenze.
3. Usa un validatore YAML per verificare la validità del tuo file. AppSpec
4. Salvare il file come `appspec.yml` nella cartella principale della versione.
5. Esegui uno dei seguenti comandi per verificare di aver inserito il AppSpec file nella directory principale:

   - Per Linux, macOS o Unix:

     ```
     find /path/to/root/directory -name appspec.yml
     ```

     Non ci sarà alcun output se il AppSpec file non viene trovato lì.

   - Per Windows:

     ```
     dir path\to\root\directory\appspec.yml
     ```

     Se il file non è memorizzato lì, verrà visualizzato un errore AppSpec File Not Found.
6. Invia la revisione ad Amazon GitHub S3 o.

   Per istruzioni, consulta Invia una revisione CodeDeploy ad Amazon S3 EC2/On-Premises (solo distribuzioni).

AppSpec modello di file per una EC2/On-Premises distribuzione con istruzioni

> **Note**
>
> Le distribuzioni su istanze di Windows Server non supportano l'elemento. `runas` Se si esegue la distribuzione su istanze di Windows Server, non includerlo nel file. AppSpec

```
# This is an appspec.yml template file for use with an EC2/On-Premises deployment in
 CodeDeploy.
# The lines in this template starting with the hashtag symbol are
#   instructional comments and can be safely left in the file or
#   ignored.
# For help completing this file, see the "AppSpec File Reference" in the
#   "CodeDeploy User Guide" at
```

```
#   https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
# Specify "os: linux" if this revision targets Amazon Linux,
#   Red Hat Enterprise Linux (RHEL), or Ubuntu Server
#   instances.
# Specify "os: windows" if this revision targets Windows Server instances.
# (You cannot specify both "os: linux" and "os: windows".)
os: linux
# os: windows
# During the Install deployment lifecycle event (which occurs between the
#   BeforeInstall and AfterInstall events), copy the specified files
#   in "source" starting from the root of the revision's file bundle
#   to "destination" on the Amazon EC2 instance.
# Specify multiple "source" and "destination" pairs if you want to copy
#   from multiple sources or to multiple destinations.
# If you are not copying any files to the Amazon EC2 instance, then remove the
#   "files" section altogether. A blank or incomplete "files" section
#   may cause associated deployments to fail.
files:
  - source:
    destination:
  - source:
    destination:
# For deployments to Amazon Linux, Ubuntu Server, or RHEL instances,
#   you can specify a "permissions"
#   section here that describes special permissions to apply to the files
#   in the "files" section as they are being copied over to
#   the Amazon EC2 instance.
#   For more information, see the documentation.
# If you are deploying to Windows Server instances,
#   then remove the
#   "permissions" section altogether. A blank or incomplete "permissions"
#   section may cause associated deployments to fail.
permissions:
  - object:
    pattern:
    except:
    owner:
    group:
    mode:
    acls:
      -
    context:
      user:
```

```
      type:
      range:
    type:
      -
# If you are not running any commands on the Amazon EC2 instance, then remove
#   the "hooks" section altogether. A blank or incomplete "hooks" section
#   may cause associated deployments to fail.
hooks:
# For each deployment lifecycle event, specify multiple "location" entries
#   if you want to run multiple scripts during that event.
# You can specify "timeout" as the number of seconds to wait until failing the
 deployment
#   if the specified scripts do not run within the specified time limit for the
#   specified event. For example, 900 seconds is 15 minutes. If not specified,
#   the default is 1800 seconds (30 minutes).
#   Note that the maximum amount of time that all scripts must finish executing
#   for each individual deployment lifecycle event is 3600 seconds (1 hour).
#   Otherwise, the deployment will stop and CodeDeploy will consider the deployment
#   to have failed to the Amazon EC2 instance. Make sure that the total number of
 seconds
#   that are specified in "timeout" for all scripts in each individual deployment
#   lifecycle event does not exceed a combined 3600 seconds (1 hour).
# For deployments to Amazon Linux, Ubuntu Server, or RHEL instances,
#   you can specify "runas" in an event to
#   run as the specified user. For more information, see the documentation.
#   If you are deploying to Windows Server instances,
#   remove "runas" altogether.
# If you do not want to run any commands during a particular deployment
#   lifecycle event, remove that event declaration altogether. Blank or
#   incomplete event declarations may cause associated deployments to fail.
# During the ApplicationStop deployment lifecycle event, run the commands
#   in the script specified in "location" starting from the root of the
#   revision's file bundle.
  ApplicationStop:
    - location:
      timeout:
      runas:
    - location:
      timeout:
      runas:
# During the BeforeInstall deployment lifecycle event, run the commands
#   in the script specified in "location".
  BeforeInstall:
    - location:
```

```
      timeout:
      runas:
    - location:
      timeout:
      runas:
# During the AfterInstall deployment lifecycle event, run the commands
#   in the script specified in "location".
  AfterInstall:
    - location:
      timeout:
      runas:
    - location:
      timeout:
      runas:
# During the ApplicationStart deployment lifecycle event, run the commands
#   in the script specified in "location".
  ApplicationStart:
    - location:
      timeout:
      runas:
    - location:
      timeout:
      runas:
# During the ValidateService deployment lifecycle event, run the commands
#   in the script specified in "location".
  ValidateService:
    - location:
      timeout:
      runas:
    - location:
      timeout:
      runas:
```

Scegli un tipo di CodeDeploy repository

La posizione di archiviazione per i file richiesta da CodeDeploy è denominata repository. L'uso di un repository dipende dalla piattaforma di elaborazione utilizzata dalla distribuzione.

- EC2/On-Premises: Per distribuire il codice dell'applicazione su una o più istanze, il codice deve essere raggruppato in un file di archivio e inserito in un repository a cui sia CodeDeploy possibile accedervi durante il processo di distribuzione. Il contenuto distribuibile e un AppSpec file

vengono raggruppati in un file di archivio, quindi caricati in uno dei tipi di repository supportati da. CodeDeploy

- AWS Lambda e Amazon ECS: le distribuzioni richiedono un AppSpec file, a cui è possibile accedere durante una distribuzione in uno dei seguenti modi:
  - Da un bucket Amazon S3.
  - Dal testo digitato direttamente nell' AppSpec editor della console. Per ulteriori informazioni, consultare Crea un AWS Implementazione della piattaforma di elaborazione Lambda (console) e Crea un'implementazione di Amazon ECS Compute Platform (console).
  - Se si utilizza il AWS CLI, è possibile fare riferimento a un AppSpec file presente sul disco rigido o su un'unità di rete. Per ulteriori informazioni, consultare Crea un AWS Implementazione della piattaforma di elaborazione Lambda (CLI) e Crea un'implementazione di Amazon ECS Compute Platform (CLI).

CodeDeploy attualmente supporta i seguenti tipi di repository:

Tipo di archivio	Dettagli repository	Piattaforme di elaborazione supportate
Amazon S3	Amazon Simple Storage Service (Amazon S3) è la AWS soluzione per lo storage di oggetti sicuro e scalabile . Amazon S3 archivia i dati come oggetti in bucket. Un oggetto è composto da un file e facoltativamente da qualsiasi metadato che lo descriva. Per archiviare un oggetto in Amazon S3, devi caricare il file in un bucket. Al caricamento del file è possibile configurare le autorizzazioni e i metadati per l'oggetto. Ulteriori informazioni:	Le distribuzioni che utilizzano le seguenti piattaforme di elaborazione possono archiviare la revisione in un bucket Amazon S3. • EC2/On-Premises • AWS Lambda • Amazon ECS

	• Crea un bucket in Amazon S3 • Invia una revisione CodeDeploy ad Amazon S3 EC2/On-Premises (solo distribuzioni) • Distribuzione automatica da Amazon S3 utilizzando CodeDeploy	
GitHub	È possibile archiviare le revisioni delle applicazioni in GitHubrepository. È possibile attivare una distribuzione da un GitHub repository ogni volta che il codice sorgente in tale repository viene modificato. Ulteriori informazioni: • Integrazione con CodeDeploy GitHub • Tutorial: utilizzare CodeDeploy per distribuire un'applicazione da GitHub	Solo EC2/On-Premises le distribuzioni possono archiviare la revisione in un repository. GitHub

Bitbucket	Puoi distribuire il codice a gruppi di distribuzione di istanze EC2 utilizzando la pipe in Bitbucket Pipelines . CodeDeploy Bitbucket Pipelines offre funzionalità di integrazione continua e distribuzione continua (), tra cui Bitbucket Deployments. CI/CD La CodeDeploy pipe invia prima l'artefatto a un bucket S3 che hai specificato, quindi distribuisce l'elemento di codice dal bucket. Ulteriori informazioni: • CodeDeploy Guarda la pipe per Bitbucket	Solo EC2/On-Premises le distribuzioni possono archiviar e la revisione in un repository. BitBucket

> **Note**
>
> Una AWS Lambda distribuzione funziona solo con un repository Amazon S3.

Invia una revisione CodeDeploy ad Amazon S3 EC2/On-Premises (solo distribuzioni)

Dopo aver pianificato la revisione come descritto in Pianifica una revisione per CodeDeploy e aver aggiunto un AppSpec file alla revisione come descritto inAggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy, sei pronto per raggruppare i file dei componenti e inviare la revisione ad Amazon S3. Per le distribuzioni su istanze Amazon EC2, dopo aver inviato la revisione, puoi CodeDeploy utilizzare per distribuire la revisione da Amazon S3 alle istanze.

> **Note**
>
> CodeDeploy può essere utilizzato anche per distribuire le revisioni che sono state trasferite a. GitHub Per ulteriori informazioni, consulta la documentazione. GitHub

Si presuppone che siano già state seguite le istruzioni in Guida introduttiva con CodeDeploy per configurare il AWS CLI. Questo è particolarmente importante per chiamare il comando push descritto più avanti.

Assicurati di avere un bucket Amazon S3. Segui le istruzioni in Creare un bucket.

Se la distribuzione è su istanze Amazon EC2, il bucket Amazon S3 di destinazione deve essere creato o esistere nella stessa regione delle istanze di destinazione. Ad esempio, se desideri distribuire una revisione in alcune istanze nella regione Stati Uniti orientali (Virginia settentrionale) e altre istanze nella regione Stati Uniti occidentali (Oregon), devi avere un bucket nella regione Stati Uniti orientali (Virginia settentrionale) con una copia della revisione e un altro bucket nella regione Stati Uniti occidentali (Oregon) con un'altra copia della stessa revisione. In questo scenario, sarebbe quindi necessario creare due distribuzioni separate, una nella regione Stati Uniti orientali (Virginia settentrionale) e l'altra nella regione Stati Uniti occidentali (Oregon), anche se la revisione è la stessa in entrambe le regioni e i bucket.

È necessario disporre delle autorizzazioni per il caricamento nel bucket Amazon S3. Puoi specificare queste autorizzazioni tramite una policy sui bucket di Amazon S3. Ad esempio, nella seguente policy sui bucket di Amazon S3, l'utilizzo del carattere wildcard (*) consente `111122223333` all' AWS account di caricare file in qualsiasi directory del bucket Amazon S3 denominata: `amzn-s3-demo-bucket`

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

```
            }
        }
    ]
}
```

Per visualizzare l'ID del tuo AWS account, consulta Finding Your account ID. AWS

Per informazioni su come generare e allegare una bucket policy di Amazon S3, consulta Esempi di policy Bucket.

L'utente che chiama il push comando deve disporre almeno delle autorizzazioni per caricare la revisione in ogni bucket Amazon S3 di destinazione. Ad esempio, la seguente policy consente all'utente di caricare le revisioni in qualsiasi punto del bucket Amazon S3 denominato: `amzn-s3-demo-bucket`

JSON

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

Per informazioni su come creare e allegare una policy IAM, consulta Working with policies.

Invia una revisione utilizzando il AWS CLI

> **Note**
>
> Il `push` comando raggruppa gli elementi dell'applicazione e un AppSpec file in una revisione. Il formato di file di questa revisione è un file ZIP compresso. Il comando non può essere

utilizzato con una distribuzione AWS Lambda o Amazon ECS perché ognuna prevede una revisione che è un file o. JSON-formatted YAML-formatted AppSpec

Chiama il comando push per aggregare e inviare la revisione per una distribuzione. I parametri sono:

- --application-name: (stringa) obbligatorio. Il nome dell'applicazione CodeDeploy da associare a questa revisione dell'applicazione.
- --s3-location: (stringa) obbligatorio. Informazioni sulla posizione della revisione dell'applicazione da caricare su Amazon S3. È necessario specificare un bucket Amazon S3 e una chiave. La chiave è il nome della revisione. CodeDeploy comprime il contenuto prima che venga caricato. Utilizzare il formato `s3://amzn-s3-demo-bucket/your-key.zip`.
- --ignore-hidden-files o --no-ignore-hidden-files: (booleano) Facoltativo. Usa il `--no-ignore-hidden-files` flag (impostazione predefinita) per raggruppare e caricare file nascosti su Amazon S3. Usa il `--ignore-hidden-files` flag per non raggruppare e caricare file nascosti su Amazon S3.
- --source (stringa) Facoltativo. La posizione del contenuto da distribuire e del AppSpec file sulla macchina di sviluppo da comprimere e caricare su Amazon S3. La posizione è specificata come un percorso relativo alla directory corrente. Se il percorso relativo non è specificato o se viene utilizzato un punto singolo per il percorso ("."), viene utilizzata la directory corrente.
- --description (stringa) Facoltativo. Un commento che riepiloga la revisione dell'applicazione. Se non viene specificata, viene utilizzata la stringa predefinita «Uploaded by AWS CLI 'time' UTC», dove 'time' è l'ora corrente del sistema in UTC (Coordinated Universal Time).

Puoi usare il AWS CLI per inviare una revisione per una distribuzione di Amazon EC2. Un comando di invio di esempio è simile a quanto segue:

In Linux, macOS o 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 .
```

Su 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 .
```

Questo comando funziona nel modo seguente:

- Associa i file aggregati con un'applicazione denominata `WordPress_App`.
- Collega una descrizione alla revisione.
- Ignora i file nascosti.
- Rinomina la revisione `WordPressApp.zip` e la invia e un bucket denominato `amzn-s3-demo-bucket`.
- Aggrega tutti i file nella directory root nella revisione.

Dopo che il push è andato a buon fine, puoi utilizzare la console AWS CLI o la CodeDeploy console per distribuire la revisione da Amazon S3. Per distribuire questa revisione con: AWS CLI

In Linux, macOS o 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
```

Su 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
```

Per ulteriori informazioni, consulta Crea una distribuzione con CodeDeploy.

Visualizza i dettagli della revisione dell'applicazione con CodeDeploy

Puoi utilizzare la CodeDeploy console AWS CLI, le o le CodeDeploy API per visualizzare i dettagli su tutte le revisioni delle applicazioni registrate nel tuo AWS account per un'applicazione specifica.

Per ulteriori informazioni sulla registrazione di una revisione, consulta Registra una revisione dell'applicazione in Amazon S3 con CodeDeploy.

Argomenti

- Visualizza i dettagli sulla revisione dell'applicazione (console)
- Visualizza i dettagli della revisione dell'applicazione (CLI)

Visualizza i dettagli sulla revisione dell'applicazione (console)

Per visualizzare i dettagli delle revisioni delle applicazioni:

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione espandere Deploy (Distribuisci) e scegliere Applications (Applicazioni).

   > **Note**
   >
   > Se non viene visualizzata alcuna voce, assicurati che sia selezionata la regione corretta. Nella barra di navigazione, nel selettore di regione, scegli una delle regioni elencate in Regione ed endpoint in. Riferimenti generali di AWS CodeDeploy è supportato solo in queste regioni.

3. Scegliere il nome dell'applicazione che contiene la revisione da visualizzare.
4. Nella pagina Application details (Dettagli applicazione), scegliere la scheda Revisions (Revisioni) e rivedere l'elenco delle revisioni registrate per l'applicazione. Selezionare una revisione, quindi scegliere View details (Visualizza dettagli).

Visualizza i dettagli della revisione dell'applicazione (CLI)

Per utilizzare AWS CLI per visualizzare la revisione di un'applicazione, chiamate il get-application-revisioncomando o il comando. list-application-revisions

> **Note**
>
> Riferimenti da GitHub applicare alle distribuzioni solo alle distribuzioni. EC2/On-Premises Le revisioni per le AWS Lambda distribuzioni non funzionano con. GitHub

Per visualizzare i dettagli sulla revisione di una singola applicazione, chiamate il comando get-application-revision, specificando:

- Nome dell'applicazione. Per ottenere il nome dell'applicazione, chiamate il comando list-applications.
- Per una revisione memorizzata in GitHub, il nome del GitHub repository e l'ID del commit che fa riferimento alla revisione dell'applicazione che è stata inserita nel repository.
- Per una revisione archiviata in Amazon S3, il nome del bucket Amazon S3 contenente la revisione, il nome e il tipo di file del file di archivio caricato e, facoltativamente, l'identificatore di versione Amazon S3 e l'ETag del file di archivio. Se l'identificatore di versione, ETag o entrambi sono stati specificati durante una chiamata a register-application-revision, devono essere specificati qui.

Per visualizzare i dettagli sulle revisioni di più applicazioni, chiamate il comando list-application-revisions, specificando:

- Nome dell'applicazione. Per ottenere il nome dell'applicazione, chiamate il comando list-applications.
- Facoltativamente, per visualizzare i dettagli solo per le revisioni delle applicazioni Amazon S3, il nome del bucket Amazon S3 contenente le revisioni.
- Facoltativamente, per visualizzare i dettagli solo per le revisioni delle applicazioni Amazon S3, una stringa di prefisso per limitare la ricerca alle revisioni delle applicazioni Amazon S3. (Se non specificato, CodeDeploy elencherà tutte le revisioni delle applicazioni Amazon S3 corrispondenti.)
- Facoltativamente, se i dettagli di revisione devono essere elencati distinguendo le revisioni di destinazione di un gruppo di distribuzione. (Se non specificato, CodeDeploy elencherà tutte le revisioni corrispondenti.)
- Facoltativamente, il nome di colonna e l'ordine in base al quale ordinare l'elenco dei dettagli delle revisioni. (Se non specificato, CodeDeploy elencherà i risultati in un ordine arbitrario.)

Puoi elencare tutte le revisioni o solo quelle archiviate in Amazon S3. Non puoi elencare solo le revisioni memorizzate in. GitHub

Registra una revisione dell'applicazione in Amazon S3 con CodeDeploy

Se hai già chiamato il comando push per inviare una revisione dell'applicazione ad Amazon S3, non è necessario registrare la revisione. Tuttavia, se carichi una revisione su Amazon S3 con altri mezzi e desideri che la revisione venga visualizzata nella console o tramite CodeDeploy AWS CLI il, segui questi passaggi per registrare prima la revisione.

Se hai inserito una revisione dell'applicazione in un GitHub repository e desideri che la revisione venga visualizzata nella CodeDeploy console o tramite il AWS CLI, devi seguire anche questi passaggi.

Puoi utilizzare solo le CodeDeploy API AWS CLI o per registrare le revisioni delle applicazioni in Amazon S3 o. GitHub

Argomenti

- Registra una revisione in Amazon S3 con CodeDeploy (CLI)
- Registrare una revisione GitHub con CodeDeploy (CLI)

Registra una revisione in Amazon S3 con CodeDeploy (CLI)

1. Carica la revisione su Amazon S3.
2. Chiama il comando register-application-revision, specificando:
   - Nome dell'applicazione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.
   - Informazioni relative alla revisione da registrare:
     - Il nome del bucket Amazon S3 che contiene la revisione.
     - Il nome e il tipo di file della revisione caricata. Per le AWS Lambda distribuzioni, la revisione è un AppSpec file scritto in JSON o YAML. Per le EC2/On-Premises distribuzioni, la revisione contiene una versione dei file sorgente che CodeDeploy verranno distribuiti sulle istanze o degli script che verranno eseguiti sulle istanze. CodeDeploy

> **Note**
>
> I formati di file di archivio tar e tar compressi (.tar e .tar.gz) non sono supportati per le istanze di Windows Server.

- (Facoltativo) L'identificatore della versione Amazon S3 della revisione. (Se l'identificatore di versione non è specificato, CodeDeploy utilizzerà la versione più recente.)
- (Facoltativo) L'ETag della revisione. (Se l'ETag non è specificato, CodeDeploy ignorerà la convalida dell'oggetto.)

- (Facoltativo) Qualsiasi descrizione che si desidera associare alla revisione.

Le informazioni su una revisione in Amazon S3 possono essere specificate nella riga di comando, utilizzando questa sintassi come parte della chiamata. register-application-revision (`versione eTag` sono opzionali).

Per un file di revisione per una EC2/On-Premises distribuzione:

```
--s3-location bucket=string,key=string,bundleType=tar|tgz|
zip,version=string,eTag=string
```

Per un file di revisione per una distribuzione AWS Lambda:

```
--s3-location bucket=string,key=string,bundleType=JSON|YAML,version=string,eTag=string
```

Registrare una revisione GitHub con CodeDeploy (CLI)

> **Note**
>
> AWS Lambda le distribuzioni non funzionano con. GitHub

1. Carica la revisione nel tuo repository. GitHub
2. Chiamate il comando register-application-revision, specificando:

   - Nome dell'applicazione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.

- Informazioni relative alla revisione da registrare:
  - Il nome GitHub dell'utente o del gruppo assegnato al repository che contiene la revisione, seguito da una barra (/), seguita dal nome del repository.
  - L'ID del commit che fa riferimento alla revisione nel repository.
- (Facoltativo) Qualsiasi descrizione che si desidera associare alla revisione.

Le informazioni su una revisione in GitHub possono essere specificate nella riga di comando, utilizzando questa sintassi come parte della chiamata: register-application-revision

```
--github-location repository=string,commitId=string
```

Utilizzo delle distribuzioni in CodeDeploy

In CodeDeploy, una distribuzione è il processo e i componenti coinvolti nel processo di installazione del contenuto su una o più istanze. Questo contenuto può essere costituito da codice, file Web e di configurazione, eseguibili, pacchetti, script e così via. CodeDeploy distribuisce il contenuto archiviato in un archivio di origine, in base alle regole di configurazione specificate.

Se si utilizza la piattaforma di EC2/On-Premises elaborazione, è possibile eseguire contemporaneamente due distribuzioni sullo stesso set di istanze.

CodeDeploy offre due opzioni di tipo di implementazione, implementazioni sul posto e distribuzioni. blue/green

- In-place distribuzione: l'applicazione su ogni istanza del gruppo di distribuzione viene interrotta, viene installata la revisione più recente dell'applicazione e la nuova versione dell'applicazione viene avviata e convalidata. È possibile utilizzare un sistema di bilanciamento del carico in modo che ogni istanza venga annullata durante la distribuzione e quindi ripristinata in servizio al termine della distribuzione. Solo le distribuzioni che utilizzano la piattaforma di EC2/On-Premises elaborazione possono utilizzare le distribuzioni sul posto. Per ulteriori informazioni sulle implementazioni sul posto, consulta. Panoramica di una distribuzione sul posto
- Blue/green implementazione: il comportamento della distribuzione dipende dalla piattaforma di elaborazione utilizzata:
  - Blue/green su una piattaforma di EC2/On-Premises elaborazione: le istanze in un gruppo di distribuzione (l'ambiente originale) vengono sostituite da un diverso set di istanze (l'ambiente sostitutivo) utilizzando questi passaggi:
    - Le istanze vengono fornite per l'ambiente sostitutivo.
    - La revisione più recente dell'applicazione viene installata sulle istanze sostitutive.
    - Si verifica un tempo di attesa opzionale per attività come il test delle applicazioni e la verifica del sistema.
    - Le istanze nell'ambiente sostitutivo vengono registrate con uno o più sistemi di bilanciamento del carico Elastic Load Balancing, causando il reindirizzamento del traffico verso di essi. Le istanze nell'ambiente originale vengono cancellate e possono essere terminate o mantenute in esecuzione per altri usi.

> **Note**
>
> Se utilizzi una piattaforma di EC2/On-Premises elaborazione, tieni presente che le blue/green distribuzioni funzionano solo con istanze Amazon EC2.

- Blue/green su una piattaforma AWS Lambda di calcolo Amazon ECS: il traffico viene spostato in incrementi in base a una configurazione di distribuzione canaria, lineare o completa.
- Blue/green implementazioni tramite CloudFormation: Il traffico viene spostato dalle risorse correnti alle risorse aggiornate come parte di un aggiornamento dello stack. CloudFormation Attualmente sono supportate solo le implementazioni ECS blue/green .

Per ulteriori informazioni sulle blue/green distribuzioni, consulta. Panoramica di una distribuzione blue/green

Per informazioni sulla distribuzione automatica da Amazon S3, consulta Distribuzione automatica da Amazon S3 utilizzando. CodeDeploy

Argomenti

- Crea una distribuzione con CodeDeploy
- Visualizza i dettagli CodeDeploy della distribuzione
- Visualizza i dati di registro per le CodeDeploy EC2/On-Premises distribuzioni
- Interrompi una distribuzione con CodeDeploy
- Ridistribuisci e ripristina una distribuzione con CodeDeploy
- Distribuisci un'applicazione in un altro AWS account
- Utilizzare l' CodeDeploy agente per convalidare un pacchetto di distribuzione su un computer locale

Crea una distribuzione con CodeDeploy

Puoi utilizzare la CodeDeploy console AWS CLI, le o le CodeDeploy API per creare una distribuzione che installi le revisioni delle applicazioni che hai già inviato ad Amazon S3 o, se la distribuzione è su una piattaforma di EC2/On-Premises elaborazione GitHub, sulle istanze di un gruppo di distribuzione.

Il processo di creazione di una distribuzione dipende dalla piattaforma di elaborazione utilizzata dalla distribuzione.

Argomenti

- Prerequisiti di distribuzione
- Crea un'implementazione di Amazon ECS Compute Platform (console)
- Crea un AWS Implementazione della piattaforma di elaborazione Lambda (console)
- Crea una distribuzione di EC2/On-Premises Compute Platform (console)
- Crea un'implementazione di Amazon ECS Compute Platform (CLI)
- Crea un AWS Implementazione della piattaforma di elaborazione Lambda (CLI)
- Crea un'implementazione di EC2/On-Premises Compute Platform (CLI)
- Crea una blue/green distribuzione Amazon ECS tramite CloudFormation

Prerequisiti di distribuzione

Assicurati di completare i seguenti passaggi prima di avviare una distribuzione.

Prerequisiti di distribuzione su un AWS Lambda piattaforma di calcolo

- Crea un'applicazione che include almeno un gruppo di distribuzione. Per informazioni, consulta Crea un'applicazione con CodeDeploy e Crea un gruppo di distribuzione con CodeDeploy.
- Prepara la revisione dell'applicazione, nota anche come AppSpec file, che specifica la versione della funzione Lambda che desideri distribuire. Il AppSpec file può anche specificare funzioni Lambda per convalidare la distribuzione. Per ulteriori informazioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.
- Se desideri utilizzare la configurazione della distribuzione personalizzata per la distribuzione, creala prima di avviare il processo di distribuzione. Per informazioni, consulta Create a Deployment Configuration.

Prerequisiti di implementazione su una piattaforma di elaborazione EC2/on-premises

- Per una distribuzione in loco, crea o configura le istanze da distribuire. Per informazioni, consulta Utilizzo delle istanze per CodeDeploy. Per una blue/green distribuzione, disponi di un gruppo Amazon EC2 Auto Scaling esistente da utilizzare come modello per il tuo ambiente sostitutivo oppure hai una o più istanze o gruppi Amazon EC2 Auto Scaling specificati come ambiente originale. Per ulteriori informazioni, consultare Tutorial: CodeDeploy Da utilizzare per distribuire un'applicazione in un gruppo di Auto Scaling e Integrazione CodeDeploy con Amazon EC2 Auto Scaling.

- Crea un'applicazione che include almeno un gruppo di distribuzione. Per informazioni, consulta Crea un'applicazione con CodeDeploy e Crea un gruppo di distribuzione con CodeDeploy.
- Prepara la revisione dell'applicazione che desideri distribuire alle istanze nel gruppo di distribuzione. Per informazioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.
- Se desideri utilizzare la configurazione della distribuzione personalizzata per la distribuzione, creala prima di avviare il processo di distribuzione. Per informazioni, consulta Create a Deployment Configuration.
- Se stai distribuendo la revisione dell'applicazione da un bucket Amazon S3, il bucket si trova nella AWS stessa regione delle istanze del gruppo di distribuzione.
- Se stai distribuendo la revisione dell'applicazione da un bucket Amazon S3, al bucket è stata applicata una policy del bucket Amazon S3. Questa policy concede alle istanze le autorizzazioni richieste per scaricare la revisione dell'applicazione.

  Ad esempio, la seguente policy sui bucket di Amazon S3 consente a qualsiasi istanza Amazon EC2 con un profilo di istanza IAM collegato contenente l'ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` di essere scaricata da qualsiasi punto del bucket Amazon S3 denominato: `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"
                ]
            }
        }
    ]
}
```

  La seguente policy sui bucket di Amazon S3 consente a qualsiasi istanza locale con un utente IAM associato contenente l'ARN di essere scaricata `arn:aws:iam::444455556666:user/`

`CodeDeployUser` da qualsiasi punto del bucket Amazon S3 denominato: `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"
                ]
            }
        }
    ]
}
```

Per informazioni su come generare e allegare una bucket policy di Amazon S3, consulta Esempi di policy Bucket.

- Se stai creando una blue/green distribuzione o hai specificato un Classic Load Balancer, Application Load Balancer o Network Load Balancer opzionale nel gruppo di distribuzione per una distribuzione locale, hai creato un VPC utilizzando Amazon VPC che contiene almeno due sottoreti. (CodeDeploy utilizza Elastic Load Balancing, che richiede che tutte le istanze di un gruppo di bilanciamento del carico si trovino in un singolo VPC.)

  Se non hai ancora creato un VPC, consulta la Amazon VPC Getting Started Guide.

- Se stai creando una blue/green distribuzione, hai configurato almeno un Classic Load Balancer, Application Load Balancer o Network Load Balancer in Elastic Load Balancing e l'hai usato per registrare le istanze che compongono l'ambiente originale.

> **Note**
>
> Le istanze nell'ambiente di sostituzione verranno registrate con il sistema di bilanciamento del carico in seguito.

Per ulteriori informazioni sulla configurazione di un sistema di bilanciamento del carico, consulta, e. Configura un sistema di bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2 Configura un sistema di bilanciamento del carico, gruppi target e listener per le distribuzioni di CodeDeploy Amazon ECS

Prerequisiti di distribuzione per una distribuzione tramite blue/green CloudFormation

- Il modello non deve necessariamente modellare le risorse per un' CodeDeploy applicazione o un gruppo di distribuzione.
- Il modello deve includere risorse per un VPC che utilizza Amazon VPC che contiene almeno due sottoreti.
- Il modello deve includere risorse per uno o più Classic Load Balancer, Application Load Balancer o Network Load Balancer in Elastic Load Balancing utilizzati per indirizzare il traffico verso i gruppi target.

Crea un'implementazione di Amazon ECS Compute Platform (console)

Questo argomento mostra come distribuire un servizio Amazon ECS utilizzando la console. Per ulteriori informazioni, consultare Tutorial: distribuire un'applicazione in Amazon ECS e Tutorial: distribuisci un servizio Amazon ECS con un test di convalida.

1. Accedi Console di gestione AWS e apri la CodeDeploy console all'indirizzo. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Esegui una delle seguenti operazioni:

   - Se desideri distribuire un'applicazione, nel riquadro di navigazione espandi Deploy (Distribuisci) e scegli Applications (Applicazioni). Seleziona il nome dell'applicazione da distribuire. Assicurati che la colonna della piattaforma Compute per la tua applicazione sia Amazon ECS.

- Se desideri ridistribuire una distribuzione, nel riquadro di navigazione espandi Deploy (Distribuisci) e scegli Deployments (Distribuzioni). Scegli la distribuzione che desideri ridistribuire e seleziona il nome della relativa applicazione nella colonna Application (Applicazione). Assicurati che la colonna della piattaforma Compute per la tua distribuzione sia Amazon ECS.

3. Nella scheda Deployments (Distribuzioni), scegli Create deployment (Crea distribuzione).

> **Note**
>
> L'applicazione deve disporre di un gruppo di distribuzione prima di poter essere distribuita. Se l'applicazione non dispone di un gruppo di distribuzione, nella scheda Gruppi di distribuzione, scegli Crea gruppo di distribuzione. Per ulteriori informazioni, consulta Crea un gruppo di distribuzione con CodeDeploy.

4. In Deployment group (Gruppo di distribuzione), scegli un gruppo di distribuzione da utilizzare per questa distribuzione.
5. Accanto a Revision location (Posizione revisione) seleziona dove si trova la revisione:

   - La mia applicazione è archiviata in Amazon S3: per informazioniSpecificare le informazioni su una revisione archiviata in un bucket Amazon S3, consulta e torna al passaggio 6.
   - Usa l' AppSpec editor: seleziona JSON o YAML, quindi inserisci il AppSpec file nell'editor. Puoi salvare il AppSpec file scegliendo Salva come file di testo. Quando fai clic su Distribuisci al termine di queste fasi ricevi un errore se JSON o YAML non è valido. Per ulteriori informazioni sulla creazione di un AppSpec file, consultaAggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy.

6. (Facoltativo) In Deployment description (Descrizione distribuzione), immetti una descrizione per questa distribuzione.
7. (Facoltativo) In Rollback configuration overrides (Rollback di sovrascritture della configurazione), puoi specificare opzioni di rollback automatiche diverse per questa distribuzione rispetto a quelle eventualmente specificate per il gruppo di distribuzione.

   Per informazioni sui rollback in CodeDeploy, vedere Ridistribuzioni e rollback della distribuzione eRidistribuisci e ripristina una distribuzione con CodeDeploy.

   Scegli tra le seguenti opzioni:

- Effettua il rollback quando una distribuzione fallisce: CodeDeploy ridistribuisce l'ultima revisione valida conosciuta come nuova distribuzione.
- Ripristino quando vengono raggiunte le soglie di allarme: se sono stati aggiunti degli allarmi al gruppo di distribuzione, CodeDeploy ridistribuisce l'ultima revisione valida conosciuta quando vengono attivati uno o più degli allarmi specificati.
- Disabilita i rollback: non eseguire i rollback per questa distribuzione.

8. Scegli Create deployment (Crea distribuzione).

   Per monitorare lo stato della distribuzione, consulta Visualizza i dettagli CodeDeploy della distribuzione.

Crea un AWS Implementazione della piattaforma di elaborazione Lambda (console)

Questo argomento mostra come distribuire una funzione Lambda utilizzando la console.

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Esegui una delle seguenti operazioni:

   - Se desideri distribuire un'applicazione, nel riquadro di navigazione espandi Deploy (Distribuisci) e scegli Applications (Applicazioni). Seleziona il nome dell'applicazione da distribuire. Assicurati che la colonna della piattaforma Compute per la tua applicazione sia AWS Lambda.
   - Se desideri ridistribuire una distribuzione, nel riquadro di navigazione espandi Deploy (Distribuisci) e scegli Deployments (Distribuzioni). Scegli la distribuzione che desideri ridistribuire e seleziona il nome della relativa applicazione nella colonna Application (Applicazione). Assicurati che la colonna della piattaforma Compute per la tua implementazione sia AWS Lambda.

3. Nella scheda Deployments (Distribuzioni), scegli Create deployment (Crea distribuzione).

> **Note**
>
> L'applicazione deve disporre di un gruppo di distribuzione prima di poter essere distribuita. Se l'applicazione non dispone di un gruppo di distribuzione, nella scheda Gruppi di distribuzione, scegli Crea gruppo di distribuzione. Per ulteriori informazioni, consulta Crea un gruppo di distribuzione con CodeDeploy.

4. In Deployment group (Gruppo di distribuzione), scegli un gruppo di distribuzione da utilizzare per questa distribuzione.
5. Accanto a Revision location (Posizione revisione) seleziona dove si trova la revisione:

   - La mia applicazione è archiviata in Amazon S3: per informazioniSpecificare le informazioni su una revisione archiviata in un bucket Amazon S3, consulta e torna al passaggio 6.
   - Usa l' AppSpec editor: seleziona JSON o YAML, quindi inserisci il AppSpec file nell'editor. Puoi salvare il AppSpec file scegliendo Salva come file di testo. Quando fai clic su Distribuisci al termine di queste fasi ricevi un errore se JSON o YAML non è valido. Per ulteriori informazioni sulla creazione di un AppSpec file, consultaAggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy.
6. (Facoltativo) In Deployment description (Descrizione distribuzione), immetti una descrizione per questa distribuzione.
7. (Facoltativo) Espandi le sostituzioni del gruppo di distribuzione per scegliere una configurazione di distribuzione per controllare il modo in cui il traffico viene spostato alla versione della funzione Lambda diversa da quella specificata nel gruppo di distribuzione.

   Per ulteriori informazioni, consulta Configurazioni di distribuzione su un AWS Lambda piattaforma di calcolo.
8. (Facoltativo) In Rollback configuration overrides (Rollback di sovrascritture della configurazione), puoi specificare opzioni di rollback automatiche diverse per questa distribuzione rispetto a quelle eventualmente specificate per il gruppo di distribuzione.

   Per informazioni sui rollback in, consulta e. CodeDeploy Ridistribuzioni e rollback della distribuzione Ridistribuisci e ripristina una distribuzione con CodeDeploy

   Scegli tra le seguenti opzioni:

   - Effettua il rollback quando una distribuzione fallisce: CodeDeploy ridistribuisce l'ultima revisione valida conosciuta come nuova distribuzione.

- Ripristino quando vengono raggiunte le soglie di allarme: se sono stati aggiunti degli allarmi al gruppo di distribuzione, CodeDeploy ridistribuisce l'ultima revisione valida conosciuta quando vengono attivati uno o più degli allarmi specificati.
- Disabilita i rollback: non eseguire i rollback per questa distribuzione.

9. Scegli Create deployment (Crea distribuzione).

   Per monitorare lo stato della distribuzione, consulta Visualizza i dettagli CodeDeploy della distribuzione.

Crea una distribuzione di EC2/On-Premises Compute Platform (console)

Questo argomento mostra come distribuire un'applicazione su un server Amazon EC2 o locale utilizzando la console.

1. Accedi Console di gestione AWS e apri la console all'indirizzo. CodeDeploy https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Esegui una delle seguenti operazioni:

   - Se desideri distribuire un'applicazione, nel riquadro di navigazione espandi Deploy (Distribuisci) e scegli Applications (Applicazioni). Seleziona il nome dell'applicazione da distribuire. Verificare che la colonna Compute platform (Piattaforma di calcolo) per l'applicazione indichi EC2/On-Premises.
   - Se desideri ridistribuire una distribuzione, nel riquadro di navigazione espandi Deploy (Distribuisci) e scegli Deployments (Distribuzioni). Individua la distribuzione che desideri ridistribuire, quindi scegli il nome della relativa applicazione nella colonna Application (Applicazione). Assicurati che la colonna Compute platform (Piattaforma di calcolo) per la distribuzione sia EC2/On-Premises.

3. Nella scheda Deployments (Distribuzioni), scegli Create deployment (Crea distribuzione).

> **Note**
>
> L'applicazione deve disporre di un gruppo di distribuzione prima di poter essere distribuita. Se l'applicazione non dispone di un gruppo di distribuzione, nella scheda Gruppi di distribuzione, scegli Crea gruppo di distribuzione. Per ulteriori informazioni, consulta Crea un gruppo di distribuzione con CodeDeploy.

4. In Deployment group (Gruppo di distribuzione), scegli un gruppo di distribuzione da utilizzare per questa distribuzione.
5. Accanto a Repository type (Tipo di repository) scegliere il tipo di repository in cui è archiviata la revisione:
   - La mia applicazione è archiviata in Amazon S3: per informazioniSpecificare le informazioni su una revisione archiviata in un bucket Amazon S3, consulta e torna al passaggio 6.
   - La mia applicazione è archiviata in GitHub: per informazioni, consultaSpecificare le informazioni su una revisione memorizzata in un repository GitHub, quindi torna al passaggio 6.
6. (Facoltativo) In Deployment description (Descrizione distribuzione), immetti una descrizione per questa distribuzione.
7. (Facoltativo) Espandi la configurazione di distribuzione Override per scegliere una configurazione di distribuzione per controllare il modo in cui il traffico viene spostato su Amazon EC2 o sul server locale diversa da quella specificata nel gruppo di distribuzione.

   Per ulteriori informazioni, consulta Utilizzo delle configurazioni di distribuzione in CodeDeploy.
8. a. Seleziona Non interrompere la distribuzione se l'evento del ApplicationStop ciclo di vita fallisce se desideri che la distribuzione su un'istanza abbia esito positivo se l'evento del ciclo di vita fallisce. `ApplicationStop`

   b. Espandi le impostazioni aggiuntive sul comportamento di distribuzione per specificare in che modo CodeDeploy gestisce i file in una posizione di destinazione di distribuzione che non facevano parte della precedente distribuzione riuscita.

      Scegli tra le seguenti opzioni:

      - Implementazione non riuscita: viene segnalato un errore e lo stato della distribuzione viene modificato in`Failed`.

- Sovrascrivi il contenuto: se nella posizione di destinazione esiste un file con lo stesso nome, la versione della revisione dell'applicazione lo sostituisce.
- Conserva il contenuto: se nella posizione di destinazione esiste un file con lo stesso nome, il file viene conservato e la versione nella revisione dell'applicazione non viene copiata nell'istanza.

Per ulteriori informazioni, consulta Comportamento di rollback con il contenuto esistente.

9. (Facoltativo) In Rollback configuration overrides (Rollback di sovrascritture della configurazione), puoi specificare opzioni di rollback automatiche diverse per questa distribuzione rispetto a quelle eventualmente specificate per il gruppo di distribuzione.

   Per informazioni sui rollback in CodeDeploy, consulta e. Ridistribuzioni e rollback di distribuzione Ridistribuisci e ripristina una distribuzione con CodeDeploy

   Scegli tra le seguenti opzioni:

   - Effettua il rollback quando una distribuzione fallisce: CodeDeploy ridistribuisce l'ultima revisione valida conosciuta come nuova distribuzione.
   - Ripristino quando vengono raggiunte le soglie di allarme: se gli allarmi sono stati aggiunti al gruppo di distribuzione, CodeDeploy implementa l'ultima revisione valida nota quando vengono attivati uno o più degli allarmi specificati.
   - Disabilita i rollback: non eseguire i rollback per questa distribuzione.
10. Selezionare Start deployment (Avvia distribuzione).

    Per monitorare lo stato della distribuzione, consulta Visualizza i dettagli CodeDeploy della distribuzione.

Argomenti

- Specificare le informazioni su una revisione archiviata in un bucket Amazon S3
- Specificare le informazioni su una revisione memorizzata in un repository GitHub

Specificare le informazioni su una revisione archiviata in un bucket Amazon S3

Se stai seguendo i passaggi indicatiCrea una distribuzione di EC2/On-Premises Compute Platform (console), segui questi passaggi per aggiungere dettagli sulla revisione di un'applicazione archiviata in un bucket Amazon S3.

1. Copia il link Amazon S3 della tua revisione nella posizione di revisione. Per individuare il valore del link:

   a. In una scheda del browser separata:

      Accedi a Console di gestione AWS e apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/

      Individua e seleziona la revisione.

   b. Se il riquadro Properties (Proprietà) non è visibile, scegli il pulsante Properties (Proprietà).

   c. Nel riquadro Proprietà, copia il valore del campo Link nella casella Posizione di revisione della CodeDeploy console.

   Per specificare un ETag (un checksum di file) come parte della posizione della revisione:

   - Se il valore del campo Link termina in **`?versionId=`*`versionId`***, aggiungi **`&etag=`** e l'ETag al termine del valore del campo Link.
   - Se il valore del campo Link non specifica un ID versione, aggiungi **`?etag=`** e l'ETag al termine del valore del campo Link.

   > **Note**
   >
   > Sebbene non sia semplice come copiare il valore del campo Link, puoi anche digitare la posizione della revisione in uno dei seguenti formati:
   > **`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`***
   > **`https://`*`bucket-name`*`.s3.amazonaws.com/`*`folders`*`/`*`objectName`***

2. Se nell'elenco File type (Tipo di file) viene visualizzato un messaggio che indica che non è possibile rilevare il tipo di file, scegli il tipo di file della revisione. In caso contrario, accetta il tipo di file rilevato.

Specificare le informazioni su una revisione memorizzata in un repository GitHub

Se stai seguendo la procedura descritta in precedenzaCrea una distribuzione di EC2/On-Premises Compute Platform (console), segui questa procedura per aggiungere dettagli sulla revisione di un'applicazione archiviata in un repository. GitHub

1. In Connect to GitHub, effettuate una delle seguenti operazioni:

   - Per creare una connessione per CodeDeploy le applicazioni a un GitHub account, in un'altra scheda del browser Web, disconnettiti da GitHub. In GitHub Account, inserisci un nome per identificare questa connessione, quindi scegli Connetti a GitHub. La pagina web ti chiede di autorizzare l'interazione con CodeDeploy GitHub la tua applicazione. Continua alla fase 2.
   - Per utilizzare una connessione già creata, in GitHubAccount, selezionane il nome, quindi scegli Connetti a GitHub. Continua alla fase 4.
   - Per creare una connessione a un altro GitHub account, in un'altra scheda del browser Web, esci da GitHub. Scegli Connetti a un altro GitHub account, quindi scegli Connetti a GitHub. Continua alla fase 2.
2. Se ti viene richiesto di accedere a GitHub, segui le istruzioni nella pagina di accesso. Accedi con il tuo nome GitHub utente o email e password.
3. Se viene visualizzata una pagina Authorize application (Autorizza applicazione), scegli Authorize application (Autorizza applicazione).
4. Nella pagina Crea distribuzione, nella casella Nome archivio, inserisci il nome GitHub dell'utente o dell'organizzazione che contiene la revisione, seguito da una barra (/), seguita dal nome del repository che contiene la revisione. In caso di dubbi sul valore da digitare:

   a. In un'altra scheda del browser Web, accedi alla dashboard. GitHub

   b. In Your repositories (I tuoi repository), passa il puntatore del mouse sul nome del repository di destinazione. Viene visualizzato un tooltip che mostra il nome dell' GitHub utente o dell'organizzazione, seguito da una barra (/), seguita dal nome del repository. Digita questo valore visualizzato nella casella Repository name (Nome repository).

   > **Note**
   >
   > Se il nome del repository di destinazione non è visibile in I tuoi repository, utilizza la GitHub casella di ricerca per trovare il nome del repository di destinazione e il nome dell'utente o dell'organizzazione. GitHub

5. In Commit ID (ID commit), digita l'ID commit che si riferisce alla revisione nel repository. In caso di dubbi sul valore da digitare:

   a. In un'altra scheda del browser Web, accedi alla dashboard. GitHub

   b. In Your repositories (I tuoi repository), sceglie il nome del repository contenente il commit di destinazione.

   c. Nell'elenco di commit, trova e copia l'ID commit che si riferisce alla revisione nel repository. Questo ID è in genere lungo 40 caratteri ed è composto da lettere e numeri. Non utilizzare la versione breve dell'ID commit, che è in genere costituita dai primi 10 caratteri della versione più lunga dell'ID commit.

   d. Incolla l'ID commit nella casella Commit ID (ID commit).

Crea un'implementazione di Amazon ECS Compute Platform (CLI)

Dopo aver creato l'applicazione e la revisione (nelle distribuzioni di Amazon ECS, questo è il file): AppSpec

Chiama il comando create-deployment, specificando:

- Un nome dell'applicazione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.
- Un nome del gruppo di distribuzione. Per visualizzare un elenco di nomi di gruppi di distribuzione, chiamate il comando list-deployment-groups.
- Le informazioni relative alla revisione da distribuire:

  Per le revisioni archiviate in Amazon S3:

  - Il nome del bucket Amazon S3 che contiene la revisione.
  - Il nome della revisione caricata.
  - (Facoltativo) L'identificatore della versione di Amazon S3 per la revisione. (Se l'identificatore di versione non è specificato, CodeDeploy utilizza la versione più recente.)
  - (Facoltativo) L'ETag per la revisione. (Se l'ETag non è specificato, CodeDeploy ignora la convalida dell'oggetto.)

  Per le revisioni archiviate in un file che non si trova in Amazon S3, sono necessari il nome e il percorso del file. Il file di revisione è stato scritto utilizzando JSON o YAML, pertanto è probabile che abbia estensione .json o .yaml.

- (Facoltativo) Una descrizione della distribuzione.

Il file di revisione può essere specificato come file caricato in un bucket Amazon S3 o come stringa. La sintassi per ciascuna quando utilizzata come parte del comando create-deployment è la seguente:

- Bucket Amazon S3:

  `version` e `eTag` sono facoltativi.

```
--s3-location bucket=string,key=string,bundleType=JSON|
YAML,version=string,eTag=string
```

- Stringa:

```
--revision '{"revisionType": "String", "string": {"content":"revision-as-string"}}'
```

> **Note**
>
> Il comando create-deployment può caricare una revisione da un file. Per ulteriori informazioni, consultare la sezione relativa al caricamento di parametri da un file.

Per i modelli AWS Lambda di revisione della distribuzione, consulta. Aggiungi un AppSpec file per un AWS Implementazione Lambda Per una revisione di esempio, consulta AppSpec Esempio di file per un AWS Implementazione Lambda.

Per monitorare lo stato della distribuzione, consulta Visualizza i dettagli CodeDeploy della distribuzione.

Crea un AWS Implementazione della piattaforma di elaborazione Lambda (CLI)

Dopo aver creato l'applicazione e la revisione (nelle distribuzioni AWS Lambda, questo è il file): AppSpec

Chiama il comando create-deployment, specificando:

- Un nome dell'applicazione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.
- Un nome del gruppo di distribuzione. Per visualizzare un elenco di nomi di gruppi di distribuzione, chiamate il comando list-deployment-groups.
- Le informazioni relative alla revisione da distribuire:

  Per le revisioni archiviate in Amazon S3:

  - Il nome del bucket Amazon S3 che contiene la revisione.
  - Il nome della revisione caricata.
  - (Facoltativo) L'identificatore della versione di Amazon S3 per la revisione. (Se l'identificatore di versione non è specificato, CodeDeploy utilizza la versione più recente.)
  - (Facoltativo) L'ETag per la revisione. (Se l'ETag non è specificato, CodeDeploy ignora la convalida dell'oggetto.)

  Per le revisioni archiviate in un file che non si trova in Amazon S3, sono necessari il nome e il percorso del file. Il file di revisione è stato scritto utilizzando JSON o YAML, pertanto è probabile che abbia estensione .json o .yaml.
- (Facoltativo) Il nome di una configurazione della distribuzione da utilizzare. Per visualizzare un elenco di configurazioni di distribuzione, chiama il comando list-deployment-configs. (Se non specificato, utilizza una configurazione di distribuzione predefinita specifica.) CodeDeploy
- (Facoltativo) Una descrizione della distribuzione.

Il file di revisione può essere specificato come file caricato in un bucket Amazon S3 o come stringa. La sintassi per ciascuna quando utilizzata come parte del comando create-deployment è la seguente:

- Bucket Amazon S3:

  `version` e `eTag` sono facoltativi.

```
--s3-location bucket=string,key=string,bundleType=JSON|
YAML,version=string,eTag=string
```

- Stringa:

```
--revision '{"revisionType": "String", "string": {"content":"revision-as-string"}}'
```

> **Note**
>
> Il comando create-deployment può caricare una revisione da un file. Per ulteriori informazioni, consultare la sezione relativa al caricamento di parametri da un file.

Per i modelli AWS Lambda di revisione della distribuzione, consulta. Aggiungi un AppSpec file per un AWS Implementazione Lambda Per una revisione di esempio, consulta AppSpec Esempio di file per un AWS Implementazione Lambda.

Per monitorare lo stato della distribuzione, consulta Visualizza i dettagli CodeDeploy della distribuzione.

Crea un'implementazione di EC2/On-Premises Compute Platform (CLI)

Per utilizzare il AWS CLI per distribuire una revisione della piattaforma di elaborazione: EC2/On-Premises

1. Dopo aver preparato le istanze, creato l'applicazione ed eseguito il push della revisione, procedi in uno dei seguenti modi:
   - Se desideri distribuire una revisione da un bucket Amazon S3, continua subito con la fase 2.
   - Se desideri distribuire una revisione da un GitHub repository, completa prima i passaggi indicati e poi continua con il passaggio 2Connect un' CodeDeploy applicazione a un GitHub repository.
2. Chiamate il comando create-deployment, specificando:
   - `--application-name`: nome di un'applicazione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.
   - `--deployment-group-name`: nome di un gruppo di distribuzione Amazon EC2. Per visualizzare un elenco di nomi di gruppi di distribuzione, chiama il comando list-deployment-groups.
   - `--revision`: Informazioni sulla revisione da distribuire:

     Per le revisioni archiviate in Amazon S3:

     - `s3Location`: il nome del bucket Amazon S3 che contiene la revisione.
     - `s3Location --> key`: il nome della revisione caricata.
     - `s3Location --> bundleType`: il tipo di file della revisione caricata.

> **Note**
>
> I formati di file di archivio tar e tar compressi (.tar e .tar.gz) non sono supportati per le istanze di Windows Server.

- `s3Location --> version`: (Facoltativo) L'identificatore della versione di Amazon S3 per la revisione. (Se l'identificatore di versione non è specificato, CodeDeploy utilizza la versione più recente.)
- `s3Location --> eTag`: (Facoltativo) L'ETag per la revisione. (Se l'ETag non è specificato, CodeDeploy salta la convalida dell'oggetto.)

Per le revisioni memorizzate in: GitHub

- `gitHubLocation --> repository`: il nome GitHub dell'utente o del gruppo assegnato al repository che contiene la revisione, seguito da una barra (`/`), seguita dal nome del repository.
- `gitHubLocation --> commitId`: L'ID di commit per la revisione.

- `--deployment-config-name`: (Facoltativo) Il nome di una configurazione di distribuzione da utilizzare. Per visualizzare un elenco di configurazioni di distribuzione, chiama il comando list-deployment-configs. (Se non specificato, utilizza una configurazione di distribuzione predefinita specifica.) CodeDeploy
- `--ignore-application-stop-failures | --no-ignore-application-stop-failures`: (Facoltativo) Se si desidera che la distribuzione su un'istanza continui fino all'evento del ciclo di vita della `BeforeInstall` distribuzione se l'evento del ciclo di vita della `ApplicationStop` distribuzione fallisce.
- `--description`: (Facoltativo) Una descrizione per la distribuzione.
- `--file-exists-behavior`: (Facoltativo) Come parte del processo di distribuzione, l' CodeDeploy agente rimuove da ogni istanza tutti i file installati dalla distribuzione più recente. Scegli cosa succede quando i file che non facevano parte di una distribuzione precedente vengono visualizzati nelle posizioni di distribuzione di destinazione.
- `--target-instances`: per le blue/green distribuzioni, informazioni sulle istanze che appartengono all'ambiente sostitutivo in una blue/green distribuzione, inclusi i nomi di uno o più gruppi Amazon EC2 Auto Scaling o la chiave di filtro dei tag, il tipo e il valore utilizzati per identificare le istanze Amazon EC2.

> **Note**
>
> Usa questa sintassi come parte della create-deployment chiamata per specificare informazioni su una revisione in Amazon S3 direttamente sulla riga di comando. (`version` e `eTag` sono facoltativi.)
>
> ```
> --s3-location bucket=string,key=string,bundleType=tar|tgz|
> zip,version=string,eTag=string
> ```
>
> Usa questa sintassi come parte della create-deployment chiamata per specificare le informazioni su una revisione GitHub direttamente sulla riga di comando:
>
> ```
> --github-location repository=string,commitId=string
> ```
>
> Per ottenere informazioni sulle revisioni che sono già state inviate, chiamate il comando list-application-revisions.

Per monitorare lo stato della distribuzione, consulta Visualizza i dettagli CodeDeploy della distribuzione.

riferimento al comando create-deployment

La struttura e le opzioni di comando seguenti sono disponibili per il `create-deployment` comando. Per ulteriori informazioni, consulta il riferimento create-deployment nel AWS CLI Command Reference.

```
create-deployment
--application-name <value>
[--deployment-group-name <value>]
[--revision <value>]
[--deployment-config-name <value>]
[--description <value>]
[--ignore-application-stop-failures | --no-ignore-application-stop-failures]
[--target-instances <value>]
[--auto-rollback-configuration <value>]
[--update-outdated-instances-only | --no-update-outdated-instances-only]
[--file-exists-behavior <value>]
[--s3-location <value>]
[--github-location <value>]
```

```
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Connect un' CodeDeploy applicazione a un GitHub repository

Prima di poter distribuire un'applicazione da un GitHub repository per la prima volta utilizzando il AWS CLI, devi prima CodeDeploy autorizzare l'interazione con per GitHub conto del tuo account. GitHub Questo passaggio deve essere completato una volta per ogni applicazione che utilizza la CodeDeploy console.

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Selezionare Applications (Applicazioni).
3. Da Applicazioni, scegli l'applicazione che desideri collegare al tuo account GitHub utente e scegli Deploy application.

   > **Note**
   >
   > Non si sta creando una distribuzione. Questo è attualmente l'unico modo per CodeDeploy autorizzare l'interazione per GitHub conto del tuo account GitHub utente.

4. Accanto a Tipo di repository, scegli La revisione della mia applicazione è archiviata. GitHub
5. Scegli Connect a GitHub.

   > **Note**
   >
   > Se vedi il link Connetti a un altro GitHub account:
   > Potresti aver già ottenuto l'autorizzazione CodeDeploy a interagire per GitHub conto di un GitHub account diverso per l'applicazione.
   > È possibile che tu abbia revocato l'autorizzazione CodeDeploy all'interazione per GitHub conto dell' GitHub account con cui hai effettuato l'accesso per tutte le applicazioni collegate a in. CodeDeploy

Per ulteriori informazioni, consulta GitHub autenticazione con applicazioni in CodeDeploy.

6. Se non hai già effettuato l'accesso GitHub, segui le istruzioni nella pagina di accesso.
7. Nella pagina Authorize application (Autorizza applicazione), scegli Authorize application (Autorizza applicazione).
8. Ora CodeDeploy che hai l'autorizzazione, scegli Annulla e continua con i passaggi indicatiCrea un'implementazione di EC2/On-Premises Compute Platform (CLI).

Crea una blue/green distribuzione Amazon ECS tramite CloudFormation

Puoi utilizzarlo AWS CloudFormation per gestire le blue/green distribuzioni di Amazon ECS tramite. CodeDeploy La distribuzione viene generata definendo le risorse verdi e blu e specificando le impostazioni di routing e stabilizzazione del traffico da utilizzare in CloudFormation. Questo argomento tratta le differenze tra le blue/green distribuzioni di Amazon ECS gestite da CodeDeploy e le distribuzioni gestite da. CloudFormation

Per istruzioni dettagliate sull'utilizzo CloudFormation per gestire le distribuzioni di Amazon ECS, consulta Automatizza le blue/green distribuzioni ECS tramite l'uso nella Guida per l'utente. blue/green CodeDeploy AWS CloudFormationAWS CloudFormation

> **Note**
>
> La gestione delle blue/green distribuzioni Amazon ECS con non CloudFormation è disponibile nella regione Asia Pacifico (Osaka).

Differenze tra le blue/green implementazioni di Amazon ECS tramite e CodeDeploy CloudFormation

Il modello di CloudFormation stack modella le risorse e l'infrastruttura relative alle attività di Amazon ECS e anche le opzioni di configurazione per le distribuzioni. Esistono quindi differenze tra le distribuzioni standard di Amazon ECS e blue/green le distribuzioni tramite cui blue/green vengono create. CloudFormation

A differenza delle blue/green distribuzioni standard di Amazon ECS, non modelli né crei manualmente quanto segue:

- Non si crea un' AWS CodeDeploy applicazione specificando un nome che rappresenti in modo univoco ciò che si desidera distribuire.
- Non si crea un AWS CodeDeploy gruppo di distribuzione.
- Non si specifica un file di specifiche dell'applicazione (AppSpec file). Le informazioni normalmente gestite con il AppSpec file, come le opzioni di configurazione ponderate o gli eventi del ciclo di vita, vengono gestite dall'hook. `AWS::CodeDeploy::BlueGreen`

In questa tabella vengono riepilogate le differenze nel flusso di lavoro di alto livello tra i tipi di distribuzione.

Funzione	Implementazioni standard blue/green	Blue/green implementazioni tramite CloudFormation
Specificare il cluster Amazon ECS, il servizio Amazon ECS, Application Load Balancer o Network Load Balancer, il listener di produzione, il listener di test e due gruppi target.	Crea un gruppo di CodeDeploy distribuzione che specifichi queste risorse.	Crea un CloudFormation modello per modellare queste risorse.
Specificare la modifica da distribuire.	Crea un' CodeDeploy applicazione.	Crea un CloudFormation modello che specifichi l'immagine del contenitore.
Specificare la definizione dell'attività di Amazon ECS, il nome del contenitore e la porta del contenitore.	Crea un AppSpec file che specifichi queste risorse.	Crea un CloudFormation modello per modellare queste risorse.
Specificare le opzioni di spostamento del traffico di distribuzione e gli hook degli eventi del ciclo di vita.	Crea un AppSpec file che specifichi queste opzioni.	Create un CloudFormation modello che utilizzi i parametri `AWS::CodeDeploy::BlueGreen` hook per specificare queste opzioni.

Funzione	Implementazioni standard blue/green	Blue/green implementazioni tramite CloudFormation
CloudWatch allarmi.	Crea un CloudWatch allarme che attiva un rollback.	Configura un CloudWatch allarme a livello di CloudFormation stack che attiva un rollback.
Rollback/redeployment.	Specificare le opzioni di rollback e ridistribuzione.	Annulla l'aggiornamento dello stack in. CloudFormation

Monitoraggio delle blue/green distribuzioni di Amazon ECS tramite CloudFormation

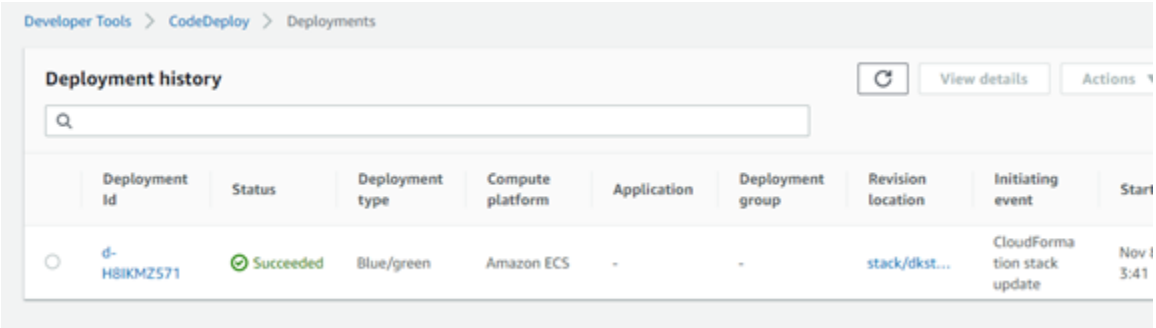
Puoi monitorare le blue/green distribuzioni tramite e. CloudFormation CodeDeploy Per informazioni sul monitoraggio tramite CloudFormation, consulta Monitoraggio blue/green degli eventi CloudFormation nella Guida per l'AWS CloudFormation utente.

Per visualizzare lo stato di implementazione delle blue/green distribuzioni in CodeDeploy

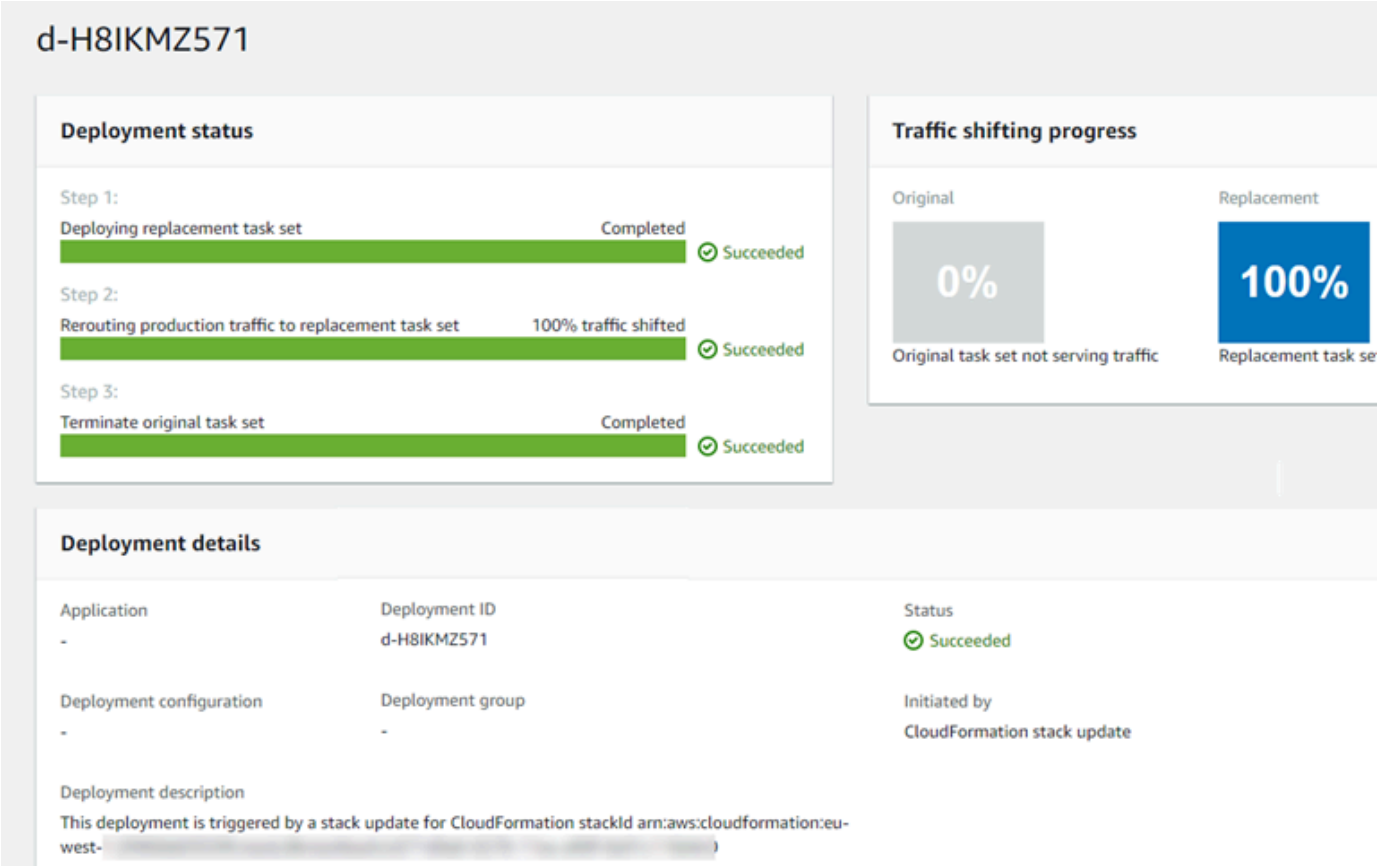
1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo. https://console.aws.amazon.com/codedeploy

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. In Deployments, viene visualizzata la distribuzione attivata dall'aggiornamento dello CloudFormation stack. Scegliere la distribuzione per visualizzare Cronologia delle distribuzioni.



3. Scegliere la distribuzione per visualizzare lo stato dello spostamento del traffico. Si noti che l'applicazione e il gruppo di distribuzione non vengono creati.



4. Quanto segue si applica per il rollback o l'arresto della distribuzione:

   - La distribuzione riuscita viene visualizzata in CodeDeploy e mostra che la distribuzione è stata avviata da. CloudFormation
   - Se si desidera interrompere e ripristinare la distribuzione, è necessario annullare l'aggiornamento dello stack in. CloudFormation

Visualizza i dettagli CodeDeploy della distribuzione

Puoi utilizzare la CodeDeploy console, le o le AWS CLI CodeDeploy API per visualizzare i dettagli sulle distribuzioni associate al tuo account. AWS

> **Note**
>
> Puoi visualizzare i log EC2/On-Premises di distribuzione delle tue istanze nelle seguenti posizioni:

- Amazon Linux, RHEL e 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

Per ulteriori informazioni, consulta Analisi dei file di log per individuare gli errori di distribuzione nelle istanze.

Argomenti

- Visualizza i dettagli di distribuzione (console)
- Visualizza i dettagli della distribuzione (CLI)

Visualizza i dettagli di distribuzione (console)

Per utilizzare la CodeDeploy console per visualizzare i dettagli della distribuzione:

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy, quindi scegli Deployments.

   > **Note**
   >
   > Se non viene visualizzata alcuna voce, assicurati che sia selezionata la regione corretta. Nella barra di navigazione, nel selettore di regione, scegli una delle regioni elencate in Regione ed endpoint in. Riferimenti generali di AWS CodeDeploy è supportato solo in queste regioni.

3. Per visualizzare ulteriori dettagli per una singola distribuzione, in Deployment history (Cronologia distribuzione), scegli l'ID della distribuzione o seleziona il pulsante accanto all'ID della distribuzione, quindi scegli View (Visualizza).

Visualizza i dettagli della distribuzione (CLI)

Per utilizzare il AWS CLI per visualizzare i dettagli della distribuzione, chiama il `get-deployment` comando o il `batch-get-deployments` comando. Puoi chiamare il comando `list-deployments` per ottenere un elenco di ID della distribuzione univoci da utilizzare come input per il comando `get-deployment` e il comando `batch-get-deployments`.

Per visualizzare i dettagli su una singola distribuzione, chiama il comando get-deployment, specificando l'identificatore di distribuzione univoco. Per ottenere l'ID di distribuzione, chiama il comando list-deployments.

Per visualizzare i dettagli su più distribuzioni, chiama il comando batch-get-deployments, specificando più identificatori di distribuzione univoci. Per ottenere gli ID di distribuzione, chiama il comando list-deployments.

Per visualizzare un elenco di ID di distribuzione, chiama il comando list-deployments, specificando:

- Il nome dell'applicazione associata alla distribuzione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.
- Il nome del gruppo di distribuzione associato alla distribuzione. Per visualizzare un elenco di nomi di gruppi di distribuzione, chiamate il comando list-deployment-groups.
- Facoltativamente, se includere i dettagli relativi alle distribuzioni in base al loro stato di distribuzione. (Se non specificato, vengono elencati tutte le distribuzioni corrispondenti, indipendentemente dallo stato di distribuzione.)
- Facoltativamente, se includere i dettagli relativi alle distribuzioni in base alle ore di inizio o fine della creazione della distribuzione, o entrambe. (Se non specificato, vengono elencate tutte le distribuzioni corrispondenti, indipendentemente dall'ora di creazione.)

Visualizza i dati di registro per le CodeDeploy EC2/On-Premises distribuzioni

Puoi visualizzare i dati di log creati da una CodeDeploy distribuzione configurando l' CloudWatch agente Amazon per visualizzare i dati aggregati nella CloudWatch console o accedendo a una singola istanza per esaminare il file di registro.

> **Note**
>
> I log non sono supportati per le distribuzioni AWS Lambda o Amazon ECS. Possono essere creati solo per le distribuzioni. EC2/On-Premises

Argomenti

- Visualizza i dati dei file di registro nella CloudWatch console Amazon
- Visualizza i file di registro su un'istanza

Visualizza i dati dei file di registro nella CloudWatch console Amazon

Quando l' CloudWatch agente Amazon viene installato su un'istanza, i dati di distribuzione per tutte le distribuzioni su quell'istanza diventano disponibili per la CloudWatch visualizzazione nella console. Per semplicità, consigliamo di utilizzarli CloudWatch per monitorare centralmente i file di registro anziché visualizzarli istanza per istanza. Per ulteriori informazioni, consulta Invia i log degli CodeDeploy agenti a CloudWatch.

Visualizza i file di registro su un'istanza

Per visualizzare la distribuzione dei dati di log per una singola istanza, è possibile effettuare l'accesso all'istanza e navigare per cercare informazioni su errori o altri eventi di distribuzione.

Argomenti

- Per visualizzare i file di registro della distribuzione su istanze di Amazon Linux, RHEL e Ubuntu Server
- Per visualizzare i file di registro della distribuzione sulle istanze di Windows Server

Per visualizzare i file di registro della distribuzione su istanze di Amazon Linux, RHEL e Ubuntu Server

Nelle istanze di Amazon Linux, RHEL e Ubuntu Server, i log di distribuzione vengono archiviati nella seguente posizione:

```
/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log
```

Per visualizzare o analizzare i log di distribuzione su istanze Amazon Linux, RHEL e Ubuntu Server, accedi all'istanza e digita il seguente comando per aprire il file di registro dell' CodeDeploy agente:

```
less /var/log/aws/codedeploy-agent/codedeploy-agent.log
```

Per la versione 2.0.x e successive, il registro dell'agente ruota ogni giorno o quando il file raggiunge i 64 MB (configurabile). Gli archivi ruotati vengono denominati `codedeploy-agent.YYYYMMDD.log` (con un suffisso numerico per le rotazioni in giornata) e vengono eliminati dopo sette giorni. La dimensione degli script log (`scripts.log`) è ruotata a 64 MB con un massimo di 8 file archiviati (configurabili).

Digitare i seguenti comandi per cercare messaggi di errore nel file di log:

Comando	Risultato
& ERROR	Mostra solo i messaggi di errore nel file di log. Utilizza un unico spazio prima e dopo la parola **ERROR**.
/ ERROR	Cerca il messaggio di errore successivo.[1]
? ERROR	Cerca il messaggio di errore precedente.[2] Usa un solo spazio prima e dopo la parola. **ERROR**
G	Vai alla fine del file di log.
g	Vai all'inizio del file di log.
q	Esci dal file di log.
h	Ulteriori informazioni sui comandi aggiuntivi.

[1] Dopo aver digitato**/ ERROR** , digita il messaggio **n** di errore successivo. Digita **N** per il precedent e messaggio di errore.

[2] Dopo aver digitato**? ERROR** , digitare **n** per il messaggio di errore successivo o digitare **N** il messaggio di errore precedente.

È inoltre possibile digitare il seguente comando per aprire un file di registro CodeDeploy degli script:

```
less /opt/codedeploy-agent/deployment-root/deployment-group-ID/deployment-ID/logs/
scripts.log
```

Digitare i seguenti comandi per cercare messaggi di errore nel file di log:

Comando	Risultato
&stderr	Mostra solo i messaggi di errore nel file di log.
/stderr	Cerca il messaggio di errore successivo.[1]
?stderr	Cerca il messaggio di errore precedente.[2]
G	Vai alla fine del file di log.
g	Vai all'inizio del file di log.
q	Esci dal file di log.
h	Ulteriori informazioni sui comandi aggiuntivi.

[1] Dopo aver digitato**/stderr**, digita il messaggio **n** di errore successivo. Digita **N** per il precedente messaggio di errore indietro.

[2] Dopo aver digitato**?stderr**, digita **n** all'indietro il messaggio di errore successivo. Digita **N** per il precedente messaggio di errore avanti.

Per visualizzare i file di registro della distribuzione sulle istanze di Windows Server

CodeDeploy file di registro dell'agente: nelle istanze di Windows Server, il file di registro dell' CodeDeploy agente viene archiviato nella seguente posizione:

Per la versione 2.0.x e successive:

```
C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log
```

Per la versione 1.8.x e precedenti:

```
C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt
```

Per visualizzare o analizzare il file di registro dell' CodeDeploy agente su un'istanza di Windows Server, accedi all'istanza e digita il comando seguente per aprire il file:

Per la versione 2.0.x e successive:

```
notepad C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log
```

Per la versione 1.8.x e precedenti:

```
notepad C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt
```

Per ricercare messaggi di errore nel file di log, premere CTRL+F, digitare **`ERROR`**, quindi premere Invio per trovare il primo errore.

CodeDeploy file di registro degli script: nelle istanze di Windows Server, i registri di distribuzione vengono archiviati nella seguente posizione:

`C:\ProgramData\Amazon\CodeDeploy\`*`deployment-group-id`*`\`*`deployment-id`*`\logs\scripts.log`

Dove:

- *`deployment-group-id`*è una stringa come `examplebf3a9c7a-7c19-4657-8684-b0c68d0cd3c4`
- *`deployment-id`*è un identificatore come `d-12EXAMPLE`

Digitate il seguente comando per aprire un file di registro CodeDeploy degli script:

```
notepad C:\ProgramData\Amazon\CodeDeploy\deployment-group-ID\deployment-ID\logs
\scripts.log
```

Per la versione 2.0.x e successive, il registro dell'agente ruota ogni giorno o quando il file raggiunge i 64 MB (configurabile). Le dimensioni del registro degli script sono ruotate a 64 MB con un massimo di 8 file archiviati (configurabili).

Per ricercare messaggi di errore nel file di log, premere CTRL+F, digitare **`stderr`**, quindi premere Invio per trovare il primo errore.

Interrompi una distribuzione con CodeDeploy

Puoi utilizzare la CodeDeploy console AWS CLI, le o le CodeDeploy API per interrompere le distribuzioni associate al tuo account. AWS

> **Warning**
>
> L'interruzione di una EC2/On-Premises distribuzione può lasciare alcune o tutte le istanze dei gruppi di distribuzione in uno stato di distribuzione indeterminato. Per ulteriori informazioni, consulta Distribuzioni interrotte e non riuscite.

È possibile arrestare una distribuzione o arrestare ed effettuare il rollback di una distribuzione.

- Arresto di una distribuzione (console)
- Arresto di una distribuzione (CLI)

> **Note**
>
> Se la distribuzione è una blue/green distribuzione tramite CloudFormation, non è possibile eseguire questa attività nella console. CodeDeploy Vai alla CloudFormation console per eseguire questa operazione.

Arresto di una distribuzione (console)

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Deploy, quindi scegli Deployments.

> **Note**
>
> Se non viene visualizzata alcuna voce, assicurati che sia selezionata la regione corretta. Nella barra di navigazione, nel selettore di regione, scegli una delle regioni elencate in Regione ed endpoint in. Riferimenti generali di AWS CodeDeploy è supportato solo in queste regioni.

3. Scegli la distribuzione che desideri arrestare con una delle seguenti operazioni:

   1. Scegli Stop deployment (Arresta distribuzione) per arrestare la distribuzione senza un rollback.
   2. Scegli Stop and roll back deployment (Arresta la distribuzione ed esegui il rollback) per arrestare la distribuzione ed eseguire un rollback.

   Per ulteriori informazioni, consulta Ridistribuisci e ripristina una distribuzione con CodeDeploy.

> **Note**
>
> Se le opzioni Stop deployment (Arresta distribuzione) e Stop and roll back deployment (Arresta distribuzione ed esegui il rollback) non sono disponibili, la distribuzione è avanzata fino a un punto in cui non può essere arrestata.

Arresto di una distribuzione (CLI)

Chiama il comando stop-deployment, specificando l'ID di distribuzione. Per visualizzare un elenco di ID di distribuzione, chiama il comando list-deployments.

Ridistribuisci e ripristina una distribuzione con CodeDeploy

CodeDeploy ripristina le distribuzioni ridistribuendo una revisione precedentemente distribuita di un'applicazione come nuova distribuzione. Queste distribuzioni ripristinate sono tecnicamente distribuzioni nuove con nuovi ID, non versioni di ripristino di una distribuzione precedente.

Le distribuzioni possono essere ripristinate automaticamente o manualmente.

Argomenti

- Rollback automatici
- Rollback manuali
- Flusso di lavoro di rollback e ridistribuzione
- Comportamento di rollback con il contenuto esistente

Rollback automatici

È possibile configurare un gruppo di distribuzione o una distribuzione in modo che venga eseguito automaticamente il ripristino quando una distribuzione non riesce o quando viene raggiunta una soglia di monitoraggio specificata. In questo caso, viene distribuita la versione più recente di una revisione di applicazione valida nota. È possibile configurare i rollback automatici al momento della creazione di un'applicazione o quando si creare o si aggiorna un gruppo di distribuzione.

Quando si crea una nuova distribuzione, è possibile anche scegliere di ignorare la configurazione di rollback automatico specificata per il gruppo di distribuzione.

> **Note**
>
> Puoi utilizzare Amazon Simple Notification Service per ricevere una notifica ogni volta che una distribuzione viene ripristinata automaticamente. Per informazioni, consulta Monitoring Deployments with Amazon SNS Event Notifications.

Per ulteriori informazioni sulla configurazione dei rollback automatici, consulta Configurazione delle opzioni avanzate per un gruppo di distribuzione.

Rollback manuali

Se non hai configurato i rollback automatici, puoi ripristinare una distribuzione manualmente creando una nuova distribuzione che utilizza una revisione dell'applicazione distribuita in precedenza, quindi seguendo la procedura per ridistribuire una revisione. Potresti dover eseguire questa operazione nel caso in cui un'applicazione si trovi in uno stato sconosciuto. Anziché dedicare molto tempo a risolvere il problema, puoi ridistribuire l'applicazione con uno stato operativo noto. Per ulteriori informazioni, consulta Crea una distribuzione con CodeDeploy.

> **Note**
>
> Se rimuovi un'istanza da un gruppo di distribuzione, CodeDeploy non disinstalla nulla che potrebbe essere già stato installato su quell'istanza.

Flusso di lavoro di rollback e ridistribuzione

Quando viene avviato il rollback automatico o quando si avvia manualmente una ridistribuzione o un rollback manuale, tenta CodeDeploy innanzitutto di rimuovere da ogni istanza partecipante tutti i file che sono stati installati con successo l'ultima volta. CodeDeploy lo fa controllando il file di pulizia:

`/opt/codedeploy-agent/deployment-root/deployment-instructions/`*`deployment-group-ID`*`-cleanup`file (per Amazon Linux, Ubuntu Server e istanze RHEL)

`C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\`*`deployment-group-ID`*`-cleanup`file (per istanze di Windows Server)

Se esiste, CodeDeploy utilizza il file di pulizia per rimuovere dall'istanza tutti i file elencati prima di iniziare la nuova distribuzione.

Ad esempio, i primi due file di testo e due file di script sono già stati distribuiti su un'istanza Amazon EC2 che esegue Windows Server e gli script hanno creato altri due file di testo durante gli eventi del ciclo di vita della distribuzione:

```
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)
```

Nel file di cleanup saranno elencati solo i primi due file di testo e due file script:

```
c:\temp\a.txt
c:\temp\b.txt
c:\temp\c.bat
c:\temp\d.bat
```

Prima della nuova distribuzione, CodeDeploy rimuoverà solo i primi due file di testo e i due file di script, lasciando intatti gli ultimi due file di testo:

```
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
```

Come parte di questo processo, CodeDeploy non cercherà di ripristinare o riconciliare in altro modo le azioni intraprese dagli script nelle distribuzioni precedenti durante le ridistribuzioni successive, che si tratti di rollback manuali o automatici. Ad esempio, se `d.bat` i file `c.bat` and contengono la logica per non ricreare i file and se già esistono, le versioni precedenti di `e.txt` e `f.txt` rimarranno invariate ogni volta che vengono eseguite e nelle distribuzioni successive. `e.txt` `f.txt` CodeDeploy `c.bat` `d.bat` È possibile aggiungere la logica a `c.bat` e `d.bat` per verificare sempre la presenza di versioni precedenti di `e.txt` e `f.txt` ed eliminarle prima di crearne di nuove.

Comportamento di rollback con il contenuto esistente

Come parte del processo di distribuzione, l' CodeDeploy agente rimuove da ogni istanza tutti i file installati nella distribuzione più recente. Se i file che non facevano parte di una distribuzione precedente vengono visualizzati nelle posizioni di distribuzione di destinazione, puoi scegliere CodeDeploy cosa farne durante la distribuzione successiva:

- Interruzione non riuscita: viene segnalato un errore e lo stato della distribuzione viene modificato in Non riuscito.
- Sovrascrivi il contenuto: la versione del file contenuta nella revisione dell'applicazione sostituisce la versione già presente sull'istanza.
- Conserva il contenuto: il file nella posizione di destinazione viene conservato e la versione nella revisione dell'applicazione non viene copiata nell'istanza.

È possibile scegliere questo comportamento quando si crea una distribuzione. Se si crea una distribuzione nella console, vedereCrea una distribuzione di EC2/On-Premises Compute Platform (console). Se si crea una distribuzione con AWS CLI, vedereCrea un'implementazione di EC2/On-Premises Compute Platform (CLI).

Puoi scegliere di mantenere i file che desideri inserire nella successiva distribuzione senza doverli aggiungere al pacchetto della revisione dell'applicazione. Ad esempio, è possibile caricare direttamente sull'istanza i file necessari per la distribuzione ma che non sono stati aggiunti al pacchetto di revisione dell'applicazione. Oppure potreste caricare file sull'istanza se le vostre applicazioni sono già nell'ambiente di produzione ma desiderate utilizzarle CodeDeploy per la prima volta per distribuirle.

Nel caso dei rollback, in cui per un errore di distribuzione viene ridistribuita la revisione dell'applicazione distribuita più recente, alla distribuzione di ripristino viene applicata l'opzione di gestione del contenuto relativa a quell'ultima distribuzione.

Tuttavia, se la distribuzione non riuscita è stata configurata per sovrascrivere, anziché conservare, i file, durante il rollback potrebbe verificarsi un risultato inaspettato. Nello specifico, potrebbe accadere che i file che devono essere conservati vengano rimossi dalla distribuzione non riuscita. I file non sono presenti nell'istanza quando viene eseguita la distribuzione di rollback.

Nell'esempio seguente, sono presenti tre distribuzioni. Tutti i file che vengono sovrascritti (eliminati) durante la seconda distribuzione non riuscita non sono più disponibili (non possono essere conservati) quando la revisione dell'applicazione 1 viene nuovamente distribuita durante la distribuzione 3:

Implementazione	Revisione dell'applicazione	Opzione sovrascrittura contenuto	Stato dell'implementazione	Comportamento e risultato
distribuzione 1	revisione dell'applicazione 1	RETAIN	Riuscito	CodeDeploy rileva i file nelle posizioni di destinazione che non sono stati distribuiti dalla distribuzione precedente. Questi file potrebbero essere stati inseriti intenzionalmente per

Implementazione	Revisione dell'applicazione	Opzione sovrascrittura contenuto	Stato dell'implementazione	Comportamento e risultato
				includerli nella distribuzione corrente. Vengono tenuti e registrati come parte del pacchetto di distribuzione corrente.
distribuzione 2	revisione dell'applicazione 2	SOVRASCRIVI	Non riuscito	Durante il processo di distribuzione, CodeDeploy elimina tutti i file che fanno parte della precedente distribuzione riuscita. Questo include i file che sono stati conservati durante la distribuzione 1. Tuttavia, la distribuzione ha esito negativo per motivi non correlati.

Implementazione	Revisione dell'applicazione	Opzione sovrascrittura contenuto	Stato dell'implementazione	Comportamento e risultato
distribuzione 3	revisione dell'applicazione 1	RETAIN		Poiché il rollback automatico è abilitato per la distribuzione o il gruppo di distribuzione, CodeDeploy distribuisce l'ultima revisione dell'applicazione notoriamente funzionante, revisione dell'applicazione 1. Tuttavia, i file che si desidera conservare nella distribuzione 1 sono stati eliminati prima del fallimento della distribuzione 2 e non possono essere recuperati da. AWS CodeDeployÈ possibile aggiungerli all'istanza, se sono necessarie per la revisione

Implementazione	Revisione dell'applicazione	Opzione sovrascrittura contenuto	Stato dell'implementazione	Comportamento e risultato
				dell'applicazione 1, oppure è possibile creare una nuova revisione dell'applicazione.

Distribuisci un'applicazione in un altro AWS account

Le organizzazioni dispongono in genere di più AWS account che utilizzano per scopi diversi (ad esempio, uno per le attività di amministrazione del sistema e un altro per le attività di sviluppo, test e produzione o uno associato agli ambienti di sviluppo e test e un altro associato all'ambiente di produzione).

Sebbene sia possibile eseguire operazioni correlate in account diversi, i gruppi di CodeDeploy distribuzione e le istanze Amazon EC2 su cui vengono distribuiti sono strettamente legati agli account con cui sono stati creati. Non è possibile, ad esempio, aggiungere un'istanza avviata con un account a un gruppo di distribuzione di un altro account.

Supponiamo di avere due AWS account: il tuo account di sviluppo e il tuo account di produzione. e che utilizzi soprattutto l'account di sviluppo. Tuttavia vorresti iniziare a eseguire distribuzioni con l'account di produzione senza la necessità di un set completo di credenziali o senza dovere effettuare la disconnessione dall'account di sviluppo per connetterti a quello di produzione.

Dopo aver seguito i passaggi di configurazione tra account, puoi avviare distribuzioni che appartengono a un altro account dell'organizzazione senza bisogno di un set completo di credenziali per quell'altro account. Puoi farlo, in parte, utilizzando una funzionalità fornita da Servizio di token di sicurezza AWS (AWS STS) che ti garantisce l'accesso temporaneo a quell'account.

Passaggio 1: crea un bucket S3 in entrambi gli account

In entrambi gli account, di sviluppo e di produzione:

- Se non l'hai già fatto, crea un bucket Amazon S3 in cui verranno archiviate le revisioni dell'applicazione per l'account di produzione. Per informazioni, consulta Creare un bucket in Amazon S3. Puoi anche usare lo stesso bucket e le stesse versioni dell'applicazione per entrambi gli account, distribuendo nell'ambiente di produzione gli stessi file testati e verificati del tuo account di sviluppo.

Fase 2: concedere le autorizzazioni del bucket Amazon S3 al profilo dell'istanza IAM dell'account di produzione

Se il bucket Amazon S3 creato nella fase 1 si trova nel tuo account di produzione, questo passaggio non è necessario. Il ruolo che assumerai in seguito sarà già dotato dell'accesso a questo bucket, in quanto anche questo si trova nell'account di produzione.

Se hai creato il bucket Amazon S3 nell'account di sviluppo, procedi come segue:

- Nell'account di produzione, crea un profilo di istanza IAM. Per informazioni, consulta Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2.

  > **Note**
  >
  > Prendi nota dell'ARN per questo profilo di istanza IAM. Dovrai aggiungerlo alla policy per più bucket che creerai successivamente.

- Nell'account di sviluppo, consenti l'accesso al bucket Amazon S3 che hai creato nell'account di sviluppo al profilo dell'istanza IAM che hai appena creato nel tuo account di produzione. Per informazioni, vedi Esempio 2: Il proprietario del bucket concede le autorizzazioni per i bucket tra account.

  Osserva quanto segue nel completare la procedura necessaria per concedere autorizzazioni per il bucket multiaccount:

  - Nella procedura guidata di esempio, l'account A rappresenta l'account di sviluppo, mentre l'account B rappresenta quello di produzione.
  - Quando esegui le attività dell'account A (di sviluppo), modifica la seguente policy del bucket per concedere le autorizzazioni multiaccount anziché utilizzare la policy di esempio fornita nella procedura guidata.

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*rappresenta il numero di account dell'account di produzione in cui hai appena creato il profilo dell'istanza IAM.

*role-name*rappresenta il nome del profilo dell'istanza IAM che hai appena creato.

*bucket-name*rappresenta il nome del bucket creato nel passaggio 1. Assicurati di includere `/*` dopo il nome del bucket per fornire l'accesso a ogni file all'interno del bucket.

Fase 3: Creare risorse e un ruolo per più account nell'account di produzione

Nell'account di produzione:

- Crea CodeDeploy le tue risorse (applicazione, gruppo di distribuzione, configurazione di distribuzione, istanze Amazon EC2, profilo dell'istanza Amazon EC2, ruolo di servizio e così via) utilizzando le istruzioni contenute in questa guida.
- Crea un ruolo aggiuntivo, un ruolo IAM multiaccount, che un utente del tuo account di sviluppo possa assumere per eseguire CodeDeploy operazioni in questo account di produzione.

Utilizza la procedura dettagliata: delega l'accesso tra AWS account utilizzando i ruoli IAM come guida per aiutarti a creare il ruolo tra account. Invece di aggiungere le autorizzazioni di esempio riportate nella procedura dettagliata al documento relativo alla policy, dovresti allegare al ruolo almeno le seguenti due policy fornite: AWS

- `AmazonS3FullAccess`: necessaria solo se il bucket S3 si trova nell'account di sviluppo. Fornisce al presunto ruolo dell'account di produzione l'accesso completo ai servizi e alle risorse Amazon S3 nell'account di sviluppo, in cui è archiviata la revisione.
- `AWSCodeDeployDeployerAccess`: consente a un utente di registrare e distribuire le revisioni.

Se desideri non solo avviare distribuzioni ma anche creare e gestire gruppi di distribuzione, aggiungi la policy `AWSCodeDeployFullAccess` anziché la policy `AWSCodeDeployDeployerAccess`. Per ulteriori informazioni sull'utilizzo delle policy gestite da IAM per concedere le autorizzazioni per le CodeDeploy attività, consulta. AWS politiche gestite (predefinite) per CodeDeploy

È possibile collegare policy aggiuntive per eseguire attività in altri servizi AWS durante l'utilizzo di questo ruolo multiaccount.

Important

Durante la creazione del ruolo IAM per più account, prendi nota dei dettagli necessari per accedere all'account di produzione.
Per utilizzare il comando Console di gestione AWS per cambiare ruolo, dovrai fornire uno dei seguenti elementi:

- Un URL per accedere all'account di produzione con le credenziali del ruolo assunto. Troverai l'URL nella pagina Review (Revisione) visualizzata alla fine del processo di creazione del ruolo multiaccount.
- Il nome del ruolo multiaccount e l'alias o il numero di ID dell'account.

Per utilizzare il AWS CLI per cambiare ruolo, dovrai fornire quanto segue:

- L'ARN del ruolo multiaccount che assumerai.

Fase 4: caricare la revisione dell'applicazione nel bucket Amazon S3

Nell'account in cui hai creato il bucket Amazon S3:

- Carica la revisione dell'applicazione nel bucket Amazon S3. Per informazioni, consulta Invia una revisione CodeDeploy ad Amazon S3 EC2/On-Premises (solo distribuzioni).

Fase 5: Assumi il ruolo di più account e distribuisci le applicazioni

Nell'account di sviluppo, puoi utilizzare AWS CLI o Console di gestione AWS per assumere il ruolo tra account diversi e avviare la distribuzione nell'account di produzione.

Per istruzioni su come utilizzare il Console di gestione AWS per cambiare ruolo e avviare le distribuzioni, consulta Passare a un ruolo () e.Console di gestione AWSCrea una distribuzione di EC2/On-Premises Compute Platform (console)

Per istruzioni su come utilizzare per assumere il AWS CLI ruolo tra account diversi e avviare le distribuzioni, consulta Switching to an IAM role () e.AWS Command Line InterfaceCrea un'implementazione di EC2/On-Premises Compute Platform (CLI)

Per ulteriori informazioni sull'assunzione di un ruolo tramite AWS STS, consulta la Guida per l'Servizio di token di sicurezza AWS utente e assume-role AssumeRolenel Command Reference.AWS CLI

Argomento correlato:

- CodeDeploy: distribuzione da un account di sviluppo a un account di produzione

Utilizzare l' CodeDeploy agente per convalidare un pacchetto di distribuzione su un computer locale

Utilizzando l' CodeDeploy agente, puoi distribuire contenuti su un'istanza a cui hai effettuato l'accesso. Ciò consente di verificare l'integrità di un file di specifiche dell'applicazione (AppSpec file) che si intende utilizzare in una distribuzione e del contenuto che si intende distribuire.

Non è necessario creare un'applicazione e un gruppo di distribuzione. Se desideri distribuire il contenuto archiviato sull'istanza locale, non hai nemmeno bisogno di un AWS account. Per i test più semplici, puoi eseguire il codedeploy-agent deploy-local comando, senza specificare alcuna opzione,

in una directory che contiene il AppSpec file e il contenuto da distribuire. Lo strumento offre opzioni per altri test case.

Convalidando un pacchetto di distribuzione in un computer locale è possibile:

- Testare l'integrità di una revisione dell'applicazione.
- Verifica il contenuto di un AppSpec file.
- Prova CodeDeploy per la prima volta con il codice dell'applicazione esistente.
- Distribuire il contenuto in modo rapido quando si è già connessi a un'istanza.

Puoi utilizzare contenuti di distribuzione archiviati sull'istanza locale o in un tipo di repository remoto supportato (bucket Amazon S3 o repository pubblici). GitHub

Prerequisiti

Prima di avviare una distribuzione locale, completa la procedura illustrata di seguito:

- Crea o usa un tipo di istanza supportato dall'agente. CodeDeploy Per informazioni, consulta Sistemi operativi supportati dall'agente CodeDeploy.
- Installa la versione 2.0.0 o successiva dell' CodeDeploy agente. Il `deploy-local` sottocomando richiede la versione 2.0.0 o successiva. Per la versione 1.8.x e precedenti, utilizzate invece il comando standalone. `codedeploy-local` Per informazioni sulla versione, consulta Cronologia delle versioni dell'agente CodeDeploy. Per informazioni sull'installazione, consulta Installa l' CodeDeploy agente.
- Se stai distribuendo i tuoi contenuti da un bucket GitHub o da un repository Amazon S3, fornisci un utente con cui utilizzarli. CodeDeploy Per informazioni, consulta Fase 1: Configurazione.
- Se stai distribuendo la revisione dell'applicazione da un bucket Amazon S3, crea un bucket Amazon S3 nella regione in cui lavori e applica una policy per i bucket Amazon S3 al bucket. Questa policy concede alle istanze le autorizzazioni richieste per scaricare la revisione dell'applicazione.

  Ad esempio, la seguente policy sui bucket di Amazon S3 consente a qualsiasi istanza Amazon EC2 con un profilo di istanza IAM collegato contenente l'ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` di essere scaricata da qualsiasi punto del bucket Amazon S3 denominato: `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"
                ]
            }
        }
    ]
}
```

La seguente policy sui bucket di Amazon S3 consente a qualsiasi istanza locale con un utente IAM associato contenente l'ARN di essere scaricata `arn:aws:iam::444455556666:user/CodeDeployUser` da qualsiasi punto del bucket Amazon S3 denominato: `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"
                ]
            }
        }
    ]
}
```

Per informazioni su come generare e allegare una bucket policy di Amazon S3, consulta Esempi di policy Bucket.

- Se stai distribuendo la revisione dell'applicazione da un bucket GitHub o da un repository Amazon S3, configura un profilo di istanza IAM e collegalo all'istanza. Per informazioni, consulta Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2, Crea un'istanza Amazon EC2 per (CodeDeploy AWS CLI o console Amazon EC2) e Crea un'istanza Amazon EC2 per (CodeDeploy CloudFormation modello).
- Se stai distribuendo i tuoi contenuti da GitHub, crea un account e un GitHub archivio pubblico. Per creare un GitHub account, consulta Partecipa. GitHub Per creare un GitHub repository, consulta Creare un repository.

> **Note**
>
> I repository privati attualmente non sono supportati. Se i tuoi contenuti sono archiviati in un GitHub repository privato, puoi scaricarli sull'istanza e utilizzare l'`--bundle-location`opzione per specificarne il percorso locale.

- Prepara il contenuto (incluso un AppSpec file) che desideri distribuire sull'istanza e inseriscilo nell'istanza locale, nel tuo bucket Amazon S3 o nel tuo repository. GitHub Per informazioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.
- Se desideri utilizzare valori diversi da quelli predefiniti per altre opzioni di configurazione, crea il file di configurazione e posizionalo nell'istanza (per istanze `/etc/codedeploy-agent/conf/codedeployagent.yml` Amazon Linux, RHEL o Ubuntu Server o `C:\ProgramData\Amazon\CodeDeploy\conf.yml` per istanze Windows Server). Per informazioni, consulta CodeDeploy riferimento alla configurazione dell'agente.

> **Note**
>
> Se utilizzi un file di configurazione su istanze Amazon Linux, RHEL o Ubuntu Server, devi:
>
> - Utilizzare le variabili `:root_dir:` e `:log_dir:` per specificare posizioni diverse da quelle predefinite per la cartella della directory dei log e la cartella principale della distribuzione.
> - Utilizzato `sudo` per eseguire i comandi CodeDeploy dell'agente.

Crea una distribuzione locale

Nell'istanza in cui desideri creare la distribuzione locale, apri una sessione terminale (istanze Amazon Linux, RHEL o Ubuntu Server) o un prompt dei comandi (Windows Server) per eseguire i comandi dello strumento.

> **Note**
>
> Il codedeploy-agent deploy-local comando utilizza il `codedeploy-agent` file binario nelle seguenti posizioni:
>
> - Su Amazon Linux, RHEL o Ubuntu Server:`/opt/codedeploy-agent/bin/codedeploy-agent`.
> - Su Windows Server:`C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe`.
>
> Per motivi di compatibilità con le versioni precedenti, il codedeploy-local comando è disponibile anche nelle stesse posizioni:
>
> - Su Amazon Linux, RHEL o Ubuntu Server, il comando si trova in`/opt/codedeploy-agent/bin/codedeploy-local`.
> - In Windows Server, il comando si trova in`C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-local.exe`.
>
> La codedeploy-local [options] sintassi continua a funzionare e accetta le stesse codedeploy-agent deploy-local opzioni di.

Sintassi del comando base

```
codedeploy-agent deploy-local [options]
```

Riepilogo

```
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>]
```

Opzioni

-l, --bundle-location

La posizione del bundle di revisione dell'applicazione. Se non specifichi un percorso, per impostazione predefinita lo strumento utilizza la directory di lavoro. Se specifichi un valore per `--bundle-location`, devi specificare anche un valore per `--type`.

Esempi di formato del percorso del bundle:

- Istanza locale di Amazon Linux, RHEL o Ubuntu Server: `/path/to/local/bundle.tgz`
- Istanza locale di Windows Server: `C:/path/to/local/bundle`
- Bucket Amazon S3: `s3://amzn-s3-demo-bucket/bundle.tar`
- GitHub archivio: `https://github.com/`*`account-name`*`/`*`repository-name`*`/`

-t, --type

Il formato del bundle di revisione dell'applicazione. I tipi supportati includono `tgz`, `tar`, `zip` e `directory`. Se non specifichi un tipo, lo strumento utilizza `directory` per impostazione predefinita. Se specifichi un valore per `--type`, devi specificare anche un valore per `--bundle-location`.

-b, --file-exists-behavior

Indica come vengono gestiti in una posizione di destinazione della distribuzione i file già esistenti che non facevano parte della distribuzione precedente. Le opzioni includono DISALLOW, OVERWRITE, RETAIN. Per ulteriori informazioni, consulta il file ExistsBehavior in AWS CodeDeploy API Reference.

-g, --deployment-group

Il percorso della cartella di destinazione del contenuto da distribuire. Se non si specifica una cartella, lo strumento ne crea una denominata default-local-deployment-group all'interno della directory principale del gruppo di distribuzione. Per ogni distribuzione locale creata, lo strumento crea una

sottodirectory all'interno di questa cartella con nomi come local-1234 (dove il numero è l'ID del processo).

-d, --deployment-group-name

Il nome del gruppo di distribuzione esposto agli script lifecycle-hook come variabile di ambiente. `DEPLOYMENT_GROUP_NAME` L'impostazione predefinita è `LocalFleet`.

-a, --application-name

Il nome dell'applicazione esposto agli script lifecycle-hook come variabile di ambiente. `APPLICATION_NAME` Il valore predefinito è la posizione del pacchetto.

-e, --events

Un set di hook di eventi override lifecycle che desideri eseguire, in ordine, anziché gli eventi elencati nel file. AppSpec È possibile specificare più hook separati da virgole. Puoi utilizzare questa opzione se:

- Vuoi eseguire un set diverso di eventi senza dover aggiornare il file. AppSpec
- Vuoi eseguire un singolo hook di eventi come eccezione al contenuto del AppSpec file, ad esempio`ApplicationStop`.

Se non specificate né DownloadBundleinstallate gli eventi nell'elenco delle sostituzioni, questi verranno eseguiti prima di tutti gli hook di eventi specificati. Se includi DownloadBundlee Install nell'elenco delle `--events` opzioni, questi devono essere preceduti solo da eventi che normalmente vengono eseguiti prima di loro nelle distribuzioni. CodeDeploy Per informazioni, consulta AppSpec sezione 'hooks'.

-c, --agent-configuration-file

Posizione di un file di configurazione da utilizzare per la distribuzione, se archiviato in un percorso diverso da quello predefinito. Un file di configurazione specifica alternative rispetto ad altri valori e comportamenti predefiniti per una distribuzione.

Per impostazione predefinita, i file di configurazione vengono archiviati in `/etc/codedeploy-agent/conf/codedeployagent.yml` (istanze Amazon Linux, RHEL o Ubuntu Server) o `C:/ProgramData/Amazon/CodeDeploy/conf.yml` (Windows Server). Per ulteriori informazioni, consulta CodeDeploy riferimento alla configurazione dell'agente.

-A, --appspec-filename

Il nome del file. AppSpec Per le distribuzioni locali, i valori accettati sono `appspec.yml` e. `appspec.yaml` Per impostazione predefinita, il AppSpec file viene chiamato. `appspec.yml`

-h, --help

Visualizza un riepilogo del contenuto della guida.

-v, --version

Visualizza il numero di versione dello strumento.

Esempi

Di seguito vengono illustrati alcuni esempi di formati di comando validi.

```
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
```

Distribuisci un pacchetto da Amazon S3:

```
codedeploy-agent deploy-local --bundle-location s3://amzn-s3-demo-bucket/bundle.tgz --
type tgz
```

```
codedeploy-agent deploy-local --bundle-location s3://amzn-s3-demo-bucket/bundle.zip?
versionId=1234&etag=47e8 --type zip --deployment-group my-deployment-group
```

Distribuisci un pacchetto da un archivio pubblico: GitHub

```
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
```

Distribuire un pacchetto specificando più eventi del ciclo di vita:

```
codedeploy-agent deploy-local --bundle-location /path/to/local/
bundle.tar --type tar --deployment-group my-deployment-group --events
 DownloadBundle,Install,ApplicationStart,HealthCheck
```

Arresta un'applicazione precedentemente distribuita utilizzando l'evento del ciclo di vita: ApplicationStop

```
codedeploy-agent deploy-local --bundle-location /path/to/local/bundle.tgz --type tgz --
deployment-group my-deployment-group --events ApplicationStop
```

Effettuare la distribuzione utilizzando un ID del gruppo di distribuzione specifico:

```
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
```

Monitoraggio delle implementazioni in CodeDeploy

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle soluzioni CodeDeploy esistenti. AWS È necessario raccogliere i dati di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica. Prima di iniziare il monitoraggio CodeDeploy, tuttavia, è necessario creare un piano di monitoraggio che includa le risposte alle seguenti domande:

- Quali sono gli obiettivi del monitoraggio?
- Di quali risorse si intende eseguire il monitoraggio?
- Con quale frequenza sarà eseguito il monitoraggio di queste risorse?
- Quali strumenti di monitoraggio verranno utilizzati?
- Chi eseguirà i processi di monitoraggio?
- Chi deve ricevere una notifica quando si verifica un problema?

Il passaggio successivo consiste nello stabilire una linea di base per CodeDeploy le normali prestazioni nell'ambiente in uso, misurando le prestazioni in diversi momenti e in diverse condizioni di carico. Durante il monitoraggio CodeDeploy, archivia i dati di monitoraggio cronologici in modo da poterli confrontare con i dati sulle prestazioni correnti, identificare i normali modelli di prestazioni e le anomalie delle prestazioni e ideare metodi per risolvere i problemi.

Ad esempio, se utilizzi CodeDeploy, puoi monitorare lo stato delle distribuzioni e delle istanze di destinazione. Quando le distribuzioni o le istanze falliscono, potrebbe essere necessario riconfigurare un file di specifiche dell'applicazione, reinstallare o aggiornare l' CodeDeploy agente, aggiornare le impostazioni in un'applicazione o in un gruppo di distribuzione o apportare modifiche alle impostazioni dell'istanza o a un file. AppSpec

Per stabilire una baseline, devi monitorare almeno gli elementi seguenti:

- Eventi e stato della distribuzione
- Eventi e stato dell'istanza

Strumenti di monitoraggio automatici

AWS fornisce diversi strumenti che è possibile utilizzare per il monitoraggio. CodeDeploy Alcuni di questi strumenti possono essere configurati in modo che eseguano automaticamente il monitoraggio,

mentre altri richiedono l'intervento manuale. Si consiglia di automatizzare il più possibile i processi di monitoraggio.

Puoi utilizzare i seguenti strumenti di monitoraggio automatizzato per osservare CodeDeploy e segnalare quando qualcosa non va:

- Amazon CloudWatch Alarms: monitora una singola metrica in un periodo di tempo specificato ed esegui una o più azioni in base al valore della metrica rispetto a una determinata soglia in diversi periodi di tempo. L'azione è una notifica inviata a un argomento di Amazon Simple Notification Service (Amazon SNS) o a una policy di Amazon EC2 Auto Scaling. CloudWatch gli allarmi non richiamano azioni semplicemente perché si trovano in uno stato particolare; lo stato deve essere cambiato e mantenuto per un determinato numero di periodi. Per ulteriori informazioni, consulta Monitoring Deployments with Amazon CloudWatch Tools.

  Per informazioni sull'aggiornamento del ruolo del servizio per l'utilizzo con il monitoraggio degli allarmi di CloudWatch , consulta Concedere CloudWatch le autorizzazioni a un ruolo di servizio CodeDeploy. Per informazioni su come aggiungere il monitoraggio degli CloudWatch allarmi alle tue CodeDeploy operazioni, consulta Crea un'applicazione con CodeDeployCrea un gruppo di distribuzione con CodeDeploy, o. Modifica le impostazioni del gruppo di distribuzione con CodeDeploy

- Amazon CloudWatch Logs: monitora, archivia e accedi ai tuoi file di registro da AWS CloudTrail o altre fonti. Per ulteriori informazioni, consulta Monitoring Log Files nella Amazon CloudWatch User Guide.

  Per informazioni sull'utilizzo della CloudWatch console per visualizzare CodeDeploy i log, consulta Visualizza i log nella CloudWatch console CodeDeploy Logs.

- Amazon CloudWatch Events: abbina gli eventi e li indirizza a una o più funzioni o stream di destinazione per apportare modifiche, acquisire informazioni sullo stato e intraprendere azioni correttive. Per ulteriori informazioni, consulta What is Amazon CloudWatch Events nella Amazon CloudWatch User Guide.

  Per informazioni sull'utilizzo di CloudWatch Events nelle tue CodeDeploy operazioni, consultaMonitoraggio delle distribuzioni con Amazon Events CloudWatch.

- AWS CloudTrail Monitoraggio dei log: condividi i file di CloudTrail registro tra account, monitora i file di registro in tempo reale inviandoli a CloudWatch Logs, scrivi applicazioni di elaborazione dei log in Java e verifica che i file di registro non siano cambiati dopo la consegna da parte di.

CloudTrail Per ulteriori informazioni, consulta Lavorare con i file di CloudTrail registro nella Guida per l'AWS CloudTrail utente.

Per informazioni sull'utilizzo di CloudTrail with CodeDeploy, vedereMonitoring Deployments.

- Amazon Simple Notification Service: configura i trigger basati sugli eventi per ricevere notifiche via SMS o e-mail su eventi di distribuzione e istanze, come il successo o l'errore. Per ulteriori informazioni, consulta Creare un argomento e Cos'è Amazon Simple Notification Service.

  Per informazioni sulla configurazione delle notifiche di Amazon SNS per CodeDeploy, consulta. Monitoring Deployments with Amazon SNS Event Notifications

Strumenti di monitoraggio manuali

Un'altra parte importante del monitoraggio CodeDeploy consiste nel monitorare manualmente gli elementi che gli CloudWatch allarmi non coprono. Le dashboard della AWS console CodeDeploy CloudWatch, e altre, forniscono una panoramica immediata dello stato dell'ambiente. AWS Ti consigliamo di controllare anche i file di registro sulle distribuzioni. CodeDeploy

- CodeDeploy la console mostra:
  - Lo stato delle distribuzioni
  - La data e l'ora dell'ultimo tentativo e dell'ultimo esito positivo di ogni distribuzione di una revisione
  - Il numero di istanze che hanno avuto esito positivo o negativo, che sono state ignorate o che sono in corso in una distribuzione
  - Lo stato delle istanze locali
  - La data e l'ora in cui la registrazione di istanze locali è stata eseguita o annullata
- CloudWatch la home page mostra:
  - Stato e allarmi attuali
  - Grafici degli allarmi e delle risorse
  - Stato di integrità dei servizi

  Inoltre, è possibile utilizzare CloudWatch per effettuare le seguenti operazioni:
  - Crea pannelli di controllo personalizzati per monitorare i servizi di interesse.
  - Crea grafici dei dati dei parametri per la risoluzione di problemi e il rilevamento di tendenze.
  - Cerca e sfoglia tutte le metriche AWS delle tue risorse

- Crea e modifica gli allarmi per ricevere le notifiche dei problemi.

Argomenti

- Monitoring Deployments with Amazon CloudWatch Tools
- Monitoring Deployments
- Monitoring Deployments with Amazon SNS Event Notifications

Monitoraggio delle implementazioni con gli strumenti di Amazon CloudWatch

Puoi monitorare le CodeDeploy distribuzioni utilizzando i seguenti CloudWatch strumenti: Amazon CloudWatch Events, CloudWatch allarmi e Amazon Logs. CloudWatch

La revisione dei log creati dall' CodeDeploy agente e dalle distribuzioni può aiutarti a risolvere le cause degli errori di distribuzione. In alternativa alla revisione dei CodeDeploy log di un'istanza alla volta, è possibile utilizzare CloudWatch Logs per monitorare tutti i log in una posizione centrale.

Per informazioni sull'utilizzo di CloudWatch allarmi ed CloudWatch eventi per monitorare le CodeDeploy distribuzioni, consulta i seguenti argomenti.

Argomenti

- Monitoraggio delle distribuzioni con allarmi in CloudWatch CodeDeploy
- Monitoraggio delle distribuzioni con Amazon Events CloudWatch

Monitoraggio delle distribuzioni con allarmi in CloudWatch CodeDeploy

Puoi creare un CloudWatch allarme per un'istanza o un gruppo Amazon EC2 Auto Scaling che utilizzi CodeDeploy nelle tue operazioni. Un allarme controlla una singola metrica in un periodo di tempo specificato ed esegue una o più azioni in base al valore della metrica rispetto a una determinata soglia in diversi periodi di tempo. CloudWatch gli allarmi richiamano azioni quando il loro stato cambia (ad esempio, da a). `OK` `ALARM`

Utilizzando la funzionalità di CloudWatch allarme nativa, puoi specificare qualsiasi azione supportata in CloudWatch caso di errore di un'istanza utilizzata in una distribuzione, come l'invio di una notifica

Amazon SNS o l'arresto, la chiusura, il riavvio o il ripristino di un'istanza. Per CodeDeploy le tue operazioni, puoi configurare un gruppo di distribuzione per interrompere una distribuzione ogni volta che viene attivato un CloudWatch allarme associato al gruppo di distribuzione.

È possibile associare fino a dieci CloudWatch allarmi a un gruppo di CodeDeploy distribuzione. Se uno degli allarmi specificati viene attivato, la distribuzione viene interrotta e passa allo stato Interrotta. Per utilizzare questa opzione, è necessario concedere CloudWatch le autorizzazioni per il proprio ruolo di CodeDeploy servizio.

Per informazioni sulla configurazione degli CloudWatch allarmi nella CloudWatch console, consulta la sezione Creazione di CloudWatch allarmi Amazon nella Amazon CloudWatch User Guide.

Per informazioni sull'associazione di un CloudWatch allarme a un gruppo di distribuzione in CodeDeploy, consulta e. Crea un gruppo di distribuzione con CodeDeploy Modifica le impostazioni del gruppo di distribuzione con CodeDeploy

Argomenti

- Concedere CloudWatch le autorizzazioni a un ruolo di servizio CodeDeploy

Concedere CloudWatch le autorizzazioni a un ruolo di servizio CodeDeploy

Prima di poter utilizzare il monitoraggio degli CloudWatch allarmi nelle distribuzioni, è necessario concedere al ruolo di servizio utilizzato nelle CodeDeploy operazioni l' CloudWatch autorizzazione ad accedere alle risorse.

Per concedere le CloudWatch autorizzazioni a un ruolo di servizio

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nella console IAM, nel pannello di navigazione, scegli Ruoli.
3. Scegli il nome del ruolo di servizio che utilizzi nelle tue AWS CodeDeploy operazioni.
4. Nella scheda Permissions (Autorizzazioni), nell'area Inline Policies (Policy inline), scegliere Create Role Policy (Crea policy ruolo).

   —oppure—

   Se il pulsante Create Role Policy (Crea policy ruolo) non è disponibile, espandere l'area Inline Policies (Policy inline), quindi scegliere click here (fai clic qui).

5. Nella pagina Set Permissions (Imposta autorizzazioni), scegli Custom Policy (Policy personalizzata), quindi scegli Select (Seleziona).
6. Nella pagina Review Policy (Riconsulta policy), nel campo Policy Name (Nome policy), digitare un nome per identificare questa policy, ad esempio `CWAlarms`.
7. Incollare quanto segue nel campo Policy Document (Documento policy):

   JSON

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

8. Scegli Apply Policy (Applica policy).

Monitoraggio delle distribuzioni con Amazon Events CloudWatch

Puoi utilizzare Amazon CloudWatch Events per rilevare e reagire ai cambiamenti nello stato di un'istanza o di una distribuzione (un «evento») nelle tue CodeDeploy operazioni. Quindi, in base alle regole che crei, CloudWatch Events invocherà una o più azioni mirate quando una distribuzione o un'istanza entra nello stato specificato in una regola. A seconda del tipo di cambiamento di stato, potrebbe essere necessario inviare notifiche, acquisire informazioni sullo stato, eseguire un'azione correttiva, avviare eventi o eseguire altre operazioni. Puoi selezionare i seguenti tipi di obiettivi quando utilizzi CloudWatch Events come parte delle tue CodeDeploy operazioni:

- AWS Lambda funzioni
- Flussi Kinesis
- Code Amazon SQS
- Built-in obiettivi (`EC2 CreateSnapshot API callEC2 RebootInstances API call`,`EC2 StopInstances API call`, `eEC2 TerminateInstances API call`)
- Argomenti di Amazon SNS

Di seguito sono riportati alcuni casi d'uso:

- Utilizza una funzione di Lambda per inoltrare una notifica a un canale Slack quando le distribuzioni hanno esito negativo.
- Effettua il push dei dati relativi a distribuzioni o istanze su un flusso Kinesis per supportare il monitoraggio completo dello stato in tempo reale.
- Utilizza le azioni di CloudWatch allarme per arrestare, terminare, riavviare o ripristinare automaticamente le istanze Amazon EC2 quando si verifica un evento di distribuzione o istanza specificato.

Il resto di questo argomento descrive la procedura di base per la creazione di una regola Events per. CloudWatch CodeDeploy Prima di creare regole di evento da utilizzare nelle CodeDeploy operazioni, tuttavia, è necessario effettuare le seguenti operazioni:

- Completa i prerequisiti CloudWatch degli eventi. Per informazioni, consulta i prerequisiti di Amazon CloudWatch Events.
- Acquisisci familiarità con eventi, regole e obiettivi in Events. CloudWatch Per ulteriori informazioni, consulta Che cos'è Amazon CloudWatch Events? e nuovi CloudWatch eventi: monitora e rispondi alle modifiche alle tue AWS risorse.
- Crea la destinazione o le destinazioni da utilizzare nelle regole degli eventi.

Per creare una regola CloudWatch relativa agli eventi per CodeDeploy:

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Events (Eventi).
3. Scegli Create rule (Crea regola), quindi, in Event selector (Selettore evento), scegli AWS CodeDeploy.
4. Specifica un tipo di dettaglio:
   - Per creare una regola che si applichi a tutte le modifiche di stato delle istanze e delle distribuzioni, scegli Any detail type (Qualsiasi tipo di dettaglio), quindi passa alla fase 6.
   - Per creare una regola applicabile solo alle istanze, scegli Tipo di dettaglio specifico, quindi scegli State-changeNotifica CodeDeploy istanza.
   - Per creare una regola che si applichi solo alle distribuzioni, scegli Tipo di dettaglio specifico, quindi scegli CodeDeploy Notifica di distribuzione. State-change

5. Specificare le modifiche di stato cui si applica la regola:

   - Per creare una regola che si applichi a tutte le modifiche di stato, scegli Any state (Qualsiasi stato).
   - Per creare una regola che si applichi solo ad alcune modifiche di stato, scegli Specific state(s) (Stati specifici), quindi scegli uno o più valori di stato dall'elenco. La tabella seguente include i valori di stato che è possibile scegliere:

Valori di stato delle distribuzioni	Valori di stato delle istanze
NON RIUSCITO	NON RIUSCITO
START	START
STOP	PRONTA
IN CODA	RIUSCITO
PRONTA	
RIUSCITO	

6. Specificate a quali CodeDeploy applicazioni si applica la regola:

   - Per creare una regola che si applichi a tutte le applicazioni, scegli Any application (Qualsiasi applicazione), quindi passa al punto 8.
   - Per creare una regola che si applichi a una sola applicazione, scegli Specific application (Applicazione specifica), quindi seleziona il nome dell'applicazione dall'elenco.

7. Specifica i gruppi di distribuzione a cui si applica la regola:

   - Per creare una regola che si applichi a tutti i gruppi di distribuzione associati all'applicazione selezionata, scegli Any deployment group (Qualsiasi gruppo di distribuzione).
   - Per creare una regola che si applichi a uno solo dei gruppi di distribuzione associati all'applicazione selezionata, scegli Specific deployment group(s) (Gruppi di distribuzione specifici), quindi seleziona il nome del gruppo di distribuzione dall'elenco.

8. Verifica la configurazione delle regole per accertarti che soddisfi i tuoi requisiti di monitoraggio di eventi.

9. Nell'area Targets (Destinazioni), scegli Add target* (Aggiungi destinazione).

10. Nell'elenco Select target type (Seleziona tipo di destinazione), scegli il tipo di destinazione da utilizzare con questa regola, quindi configura le eventuali altre opzioni richieste da quel tipo.
11. Scegli Configura dettagli.
12. Nella pagina Configura i dettagli della regola, digita un nome e una descrizione per la regola, quindi seleziona la casella Stato per abilitare subito la regola.
13. Se la regola ti soddisfa, scegli Create rule (Crea regola).

Monitoraggio delle implementazioni con AWS CloudTrail

CodeDeploy è integrato con CloudTrail, un servizio che acquisisce le chiamate API effettuate da o per conto del CodeDeploy tuo AWS account e invia i file di registro a un bucket Amazon S3 da te specificato. CloudTrailacquisisce le chiamate API dalla CodeDeploy console, dai CodeDeploy comandi tramite o direttamente AWS CLI dalle API. CodeDeploy Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare a quale richiesta è stata effettuata CodeDeploy, l'indirizzo IP di origine da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e così via. Per ulteriori informazioni CloudTrail, incluso come configurarlo e abilitarlo, consulta la Guida AWS CloudTrail per l'utente.

CodeDeploy informazioni in CloudTrail

Quando CloudTrail la registrazione è abilitata nel tuo AWS account, le chiamate API effettuate alle CodeDeploy azioni vengono tracciate nei file di registro. CodeDeploy i record vengono scritti insieme ad altri record AWS di servizio in un file di registro. CloudTrail determina quando creare e scrivere su un nuovo file in base a un periodo di tempo e alle dimensioni del file.

Tutte le CodeDeploy azioni vengono registrate e documentate nel AWS CodeDeploy Command Line Reference e nel AWS CodeDeploy API Reference. Ad esempio, le chiamate per creare distribuzioni, eliminare applicazioni e registrare le revisioni delle applicazioni generano voci nei file di registro. CloudTrail

Ogni voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni sull'identità dell'utente contenute nel registro consentono di determinare se la richiesta è stata effettuata con credenziali utente o root, con credenziali di sicurezza temporanee per un ruolo o un utente federato o da un altro servizio. AWS Per ulteriori informazioni, consulta il campo userIdentity nel riferimento agli eventi CloudTrail .

È possibile archiviare i file di log nel bucket per un periodo di tempo indeterminato, ma è anche possibile definire regole per il ciclo di vita di Amazon S3 per archiviare o eliminare automaticamente i

file di log. Per impostazione predefinita, la crittografia lato server (SSE) di Amazon S3 viene utilizzata per crittografare i file di registro.

Puoi fare in modo che le notifiche di Amazon SNS vengano CloudTrail pubblicate quando vengono consegnati nuovi file di log. Per ulteriori informazioni, consulta l'argomento relativo alla configurazione delle notifiche Amazon SNS per CloudTrail.

Puoi anche aggregare i file di CodeDeploy log di più AWS regioni e più AWS account in un unico bucket Amazon S3. Per ulteriori informazioni, consulta Ricezione di file di CloudTrail registro da più regioni.

Comprendere CodeDeploy le voci dei file di registro

CloudTrail i file di registro possono contenere una o più voci di registro in cui ogni voce è composta da più JSON-formatted eventi. Una voce di log rappresenta una singola richiesta emessa da qualsiasi origine e include informazioni sull'operazione richiesta, eventuali parametri, la data e l'ora dell'operazione e così via. Le voci di log non vengono necessariamente create in un ordine specifico, ovvero non costituiscono una traccia di stack ordinata delle chiamate API pubbliche.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'azione di CodeDeploy creazione di un gruppo di distribuzione:

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

Monitoraggio delle distribuzioni con le notifiche degli eventi di Amazon SNS

Puoi aggiungere trigger a un gruppo di CodeDeploy distribuzione per ricevere notifiche sugli eventi relativi alle distribuzioni o alle istanze in quel gruppo di distribuzione. Queste notifiche vengono inviate ai destinatari abbonati a un argomento di Amazon SNS che hai inserito nell'azione del trigger.

Puoi ricevere notifiche per CodeDeploy eventi in messaggi SMS o e-mail. Puoi anche utilizzare i dati JSON creati quando si verifica un evento specifico in altri modi, ad esempio inviando messaggi alle

code di Amazon SQS o richiamando una funzione in. AWS Lambda Per esaminare la struttura dei dati JSON forniti per i trigger di distribuzione e istanza, consulta Formati CodeDeploy di dati JSON per i trigger.

Utilizzare i trigger per ricevere le notifiche può risultare utile per:

- Sviluppatori che hanno necessità di sapere quando una distribuzione non riesce o viene interrotta per potersi occupare della risoluzione del problema.
- Sei un amministratore di sistema che deve sapere quante istanze falliscono per monitorare lo stato della tua flotta Amazon EC2.
- Manager che desiderano conoscere il numero di eventi di istanza e distribuzione, ottenibile tramite le regole di filtraggio che instradano i diversi tipi di notifica in cartelle nel client e-mail per desktop.

Puoi creare fino a 10 trigger per ogni gruppo di CodeDeploy distribuzione, per uno qualsiasi dei seguenti tipi di eventi.

Eventi di distribuzione	Eventi di istanza
• Completato • Errore • Avviato • Arrestato • Rollback • Pronto[1] • Tutti gli eventi di distribuzione	• Completato • Errore • Avviato • Pronto[1] • Tutti gli eventi di istanza

[1] Si applica solo alle distribuzioni. blue/green Indica che nelle istanze in un ambiente di sostituzione è stata installata l'ultima revisione dell'applicazione e il traffico dall'ambiente originale può ora essere reindirizzato in un sistema di bilanciamento del carico. Per ulteriori informazioni, consulta Utilizzo delle distribuzioni in CodeDeploy.

Argomenti

- Concedi le autorizzazioni Amazon SNS per un ruolo di servizio CodeDeploy
- Crea un trigger per un evento CodeDeploy
- Modifica un trigger in un gruppo di distribuzione CodeDeploy

- Eliminare un trigger da un gruppo di distribuzione CodeDeploy
- Formati CodeDeploy di dati JSON per i trigger

Concedi le autorizzazioni Amazon SNS per un ruolo di servizio CodeDeploy

Prima che i trigger possano generare notifiche, al ruolo di servizio che utilizzi nelle tue CodeDeploy operazioni deve essere concessa l'autorizzazione per accedere alle risorse Amazon SNS.

Per concedere le autorizzazioni di Amazon SNS a un ruolo di servizio

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nella console IAM, nel pannello di navigazione, scegli Ruoli.
3. Selezionare il nome del ruolo del servizio utilizzato nelle operazioni AWS CodeDeploy .
4. Nella scheda Permissions (Autorizzazioni), nell'area Inline Policies (Policy inline), scegliere Create Role Policy (Crea policy ruolo).

   —oppure—

   Se il pulsante Create Role Policy (Crea policy ruolo) non è disponibile, espandere l'area Inline Policies (Policy inline), quindi scegliere click here (fai clic qui).
5. Nella pagina Set Permissions (Imposta autorizzazioni), scegli Custom Policy (Policy personalizzata), quindi scegli Select (Seleziona).
6. Nella pagina Review Policy (Rivedi policy), nel campo Policy Name (Nome policy), immettere un nome per questa policy, ad esempio `SNSPublish`.
7. Incollare quanto segue nel campo Policy Document (Documento policy):

   JSON

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

```
    ]
}
```

8. Scegli Apply Policy (Applica policy).

Crea un trigger per un evento CodeDeploy

Puoi creare un trigger che pubblica un argomento di Amazon Simple Notification Service (Amazon SNS) per un AWS CodeDeploy evento di distribuzione o istanza. Quindi, quando si verifica l'evento, tutti gli abbonati all'argomento associato ricevono notifiche tramite l'endpoint specificato nell'argomento, ad esempio un messaggio SMS o un messaggio e-mail. Amazon SNS offre diversi metodi per iscriversi agli argomenti.

Prima di creare un trigger, devi impostare l'argomento Amazon SNS a cui il trigger faccia riferimento. Per informazioni, consulta Creare un argomento. Quando crei un argomento, ti consigliamo di assegnargli un nome che ne identifichi lo scopo, in formati come `Topic-group-us-west-3-deploy-fail` o`Topic-group-project-2-instance-stop`.

È inoltre necessario concedere ad Amazon SNS le autorizzazioni per un ruolo di CodeDeploy servizio prima di poter inviare notifiche per il trigger. Per informazioni, consulta Concedi le autorizzazioni Amazon SNS per un ruolo di servizio CodeDeploy.

Dopo aver creato l'argomento, si possono aggiungere i sottoscrittori. Per informazioni sulla creazione, la gestione e l'iscrizione agli argomenti, consulta Cos'è Amazon Simple Notification Service.

Crea un trigger per inviare notifiche per CodeDeploy eventi (console)

È possibile utilizzare la CodeDeploy console per creare trigger per un CodeDeploy evento. Al termine del processo di installazione, viene inviato un messaggio di notifica di prova per verificare che le autorizzazioni e i dettagli dei trigger siano impostati correttamente.

Per creare un trigger per un evento CodeDeploy

1. In Console di gestione AWS, apri la AWS CodeDeploy console.
2. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

3. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
4. Nella pagina Applications (Applicazioni) scegliere il nome dell'applicazione associata al gruppo di distribuzione in cui si intende aggiungere un trigger.
5. Nella pagina Application details (Dettagli applicazione) scegliere il gruppo di distribuzione in cui si intende aggiungere un trigger.
6. Scegli Modifica.
7. Espandere Advanced - optional (Opzioni avanzate - opzionale).
8. Nell'area Triggers (Trigger), scegliere Create trigger (Crea trigger).
9. Nel riquadro Create deployment trigger (Crea trigger di distribuzione) eseguire le operazioni seguenti:

    a. In Trigger name (Nome trigger), immettere un nome per il trigger che consenta di identificarne lo scopo. Si consiglia di utilizzare formati come `Trigger-group-us-west-3-deploy-fail` o `Trigger-group-eu-central-instance-stop`.

    b. In Eventi, scegli il tipo o i tipi di eventi per attivare l'argomento Amazon SNS per l'invio di notifiche.

    c. Negli argomenti di Amazon SNS, scegli il nome dell'argomento che hai creato per l'invio di notifiche per questo trigger.

    d. Scegli Crea trigger. CodeDeploy invia una notifica di prova per confermare che l'accesso è stato configurato correttamente tra CodeDeploy e l'argomento Amazon SNS. La conferma verrà inviata in un messaggio SMS o in un messaggio e-mail, in base al tipo di endpoint selezionato per l'argomento e a seconda che sia stata effettuata la sottoscrizione all'argomento.

10. Scegli Save changes (Salva modifiche).

Crea un trigger per inviare notifiche per CodeDeploy gli eventi (CLI)

È possibile utilizzare l'interfaccia a riga di comando per includere i trigger al momento della creazione di un gruppo di distribuzione oppure è possibile aggiungere trigger a un gruppo di distribuzione esistente.

Per creare un trigger per l'invio di notifiche per un nuovo gruppo di distribuzione

Creare un file JSON per configurare il gruppo di distribuzione, quindi eseguire il comando create-deployment-group tramite l'opzione `--cli-input-json`.

Il modo più semplice per creare il file JSON consiste nell'utilizzare l'opzione `--generate-cli-skeleton` per ottenere una copia del formato JSON, quindi fornire i valori richiesti in un editor di testo normale.

1. Eseguire il comando seguente, quindi copiare i risultati in un editor di testo normale.

```
aws deploy create-deployment-group --generate-cli-skeleton
```

2. Aggiungi il nome di un' CodeDeploy applicazione esistente all'output:

```
{
    "applicationName": "TestApp-us-east-2",
    "deploymentGroupName": "",
    "deploymentConfigName": "",
    "ec2TagFilters": [
        {
            "Key": "",
            "Value": "",
            "Type": ""
        }
    ],
    "onPremisesInstanceTagFilters": [
        {
            "Key": "",
            "Value": "",
            "Type": ""
        }
    ],
    "autoScalingGroups": [
        ""
    ],
    "serviceRoleArn": "",
    "triggerConfigurations": [
        {
            "triggerName": "",
            "triggerTargetArn": "",
            "triggerEvents": [
                ""
```

```
            ]
        }
    ]
}
```

3. Fornire i valori per i parametri che si desidera configurare.

   Quando si utilizza il comando create-deployment-group, è necessario fornire i valori almeno per i seguenti parametri:

   - `applicationName`: il nome di un'applicazione già creata nel proprio account.
   - `deploymentGroupName`: un nome per il gruppo di distribuzione da creare.
   - `serviceRoleArn`: L'ARN di un ruolo di servizio esistente impostato CodeDeploy nel tuo account. Per informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.

   Nella sezione `triggerConfigurations` fornire i valori per i seguenti parametri:

   - `triggerName`: il nome da assegnare al trigger per poterlo identificare facilmente. Si consiglia di utilizzare formati come `Trigger-group-us-west-3-deploy-fail` o `Trigger-group-eu-central-instance-stop`.
   - `triggerTargetArn`: L'ARN dell'argomento Amazon SNS che hai creato per associarlo al tuo trigger, in questo formato:. `arn:aws:sns:us-east-2:444455556666:NewTestTopic`
   - `triggerEvents`: il tipo di evento o gli eventi per cui si desidera attivare le notifiche. È possibile specificare uno o più tipi di evento utilizzando le virgole per separare i diversi nomi (ad esempio, `"triggerEvents": ["DeploymentSuccess","DeploymentFailure","InstanceFailure"]`). Quando si aggiungono più tipi di evento, le notifiche vengono inviate all'argomento specificato per tutti i tipi di evento, anziché a un argomento diverso per ciascuno di essi. Puoi scegliere tra i seguenti tipi di evento:
     - DeploymentStart
     - DeploymentSuccess
     - DeploymentFailure
     - DeploymentStop
     - DeploymentRollback
     - DeploymentReady (Si applica solo alle istanze sostitutive in una distribuzione) blue/green
     - InstanceStart

- InstanceSuccess
- InstanceFailure
- InstanceReady (Si applica solo alle istanze sostitutive in una blue/green distribuzione)

L'esempio di configurazione seguente crea un gruppo di distribuzione denominato `dep-group-ghi-789-2` per un'applicazione denominata `TestApp-us-east-2` e un trigger che richiede l'invio di notifiche ogni volta che viene avviata una distribuzione, indipendentemente dall'esito:

```
{
    "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. Salva gli aggiornamenti come file JSON e chiama il file utilizzando l'opzione `--cli-input-json` quando esegui il comando create-deployment-group:

> **Important**
>
> Assicurarsi di includere `file://` prima del nome del file. Questo è obbligatorio in questo comando.

```
aws deploy create-deployment-group --cli-input-json file://filename.json
```

Al termine del processo di creazione riceverai un messaggio di notifica di prova che indica che i dettagli delle autorizzazioni e dei trigger sono impostati correttamente.

Per creare un trigger per l'invio di notifiche per un gruppo di distribuzione esistente

Per utilizzare l'opzione AWS CLI per aggiungere trigger per CodeDeploy eventi a un gruppo di distribuzione esistente, crea un file JSON per aggiornare il gruppo di distribuzione, quindi esegui il comando update-deployment-group utilizzando l'opzione. `--cli-input-json`

Il modo più semplice per creare il file JSON consiste nell'eseguire il comando get-deployment-group per ottenere una copia della configurazione del gruppo di distribuzione in formato JSON, quindi aggiornare i valori di parametro in un editor di testo normale.

1. Eseguire il comando seguente, quindi copiare i risultati in un editor di testo normale.

   ```
   aws deploy get-deployment-group --application-name application --deployment-group-name deployment-group
   ```

2. Eliminare quanto segue dall'output:

   - Eliminare `{ "deploymentGroupInfo":` all'inizio dell'output.
   - Eliminare `}` alla fine dell'output.
   - Eliminare la riga contenente `deploymentGroupId`.
   - Eliminare la riga contenente `deploymentGroupName`.

   Il contenuto del file di testo dovrebbe essere analogo al seguente:

   ```
   {
       "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. Nella sezione `triggerConfigurations` aggiungere i dati per i parametri `triggerEvents`, `triggerTargetArn` e `triggerName`. Per informazioni sui parametri di configurazione dei trigger, vedere. TriggerConfig

   Il contenuto del file di testo dovrebbe essere analogo al seguente. Questo codice richiede l'invio di notifiche ogni volta che viene avviata una distribuzione, indipendentemente dall'esito.

```
{
    "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. Salvare gli aggiornamenti come file JSON, quindi eseguire il comando update-deployment-group tramite l'opzione `--cli-input-json`. Assicurati di includere l'`--current-deployment-group-name`opzione e di sostituire il nome del tuo file JSON con: *`filename`*

   > **Important**
   >
   > Assicurarsi di includere `file://` prima del nome del file. Questo è obbligatorio in questo comando.

   ```
   aws deploy update-deployment-group --current-deployment-group-name deployment-group-name --cli-input-json file://filename.json
   ```

   Al termine del processo di creazione riceverai un messaggio di notifica di prova che indica che i dettagli delle autorizzazioni e dei trigger sono impostati correttamente.

Modifica un trigger in un gruppo di distribuzione CodeDeploy

Se i requisiti di notifica cambiano, puoi modificare il trigger anziché crearne uno nuovo.

CodeDeploy Modificare un trigger (CLI)

Per utilizzare il comando AWS CLI per modificare i dettagli dei trigger per CodeDeploy gli eventi quando aggiorni un gruppo di distribuzione, crea un file JSON per definire le modifiche alle proprietà del gruppo di distribuzione, quindi esegui il comando update-deployment-group con l'opzione. `--cli-input-json`

Il modo più semplice per creare il file JSON consiste nell'eseguire il comando get-deployment-group per ottenere i dettagli correnti del gruppo di distribuzione in formato JSON, quindi modificare i valori richiesti in un editor di testo normale.

1. Esegui il comando seguente, sostituendo i nomi dell'applicazione e del gruppo di distribuzione con e: *`application deployment-group`*

```
aws deploy get-deployment-group --application-name application --deployment-group-
name deployment-group
```

2. Copiare i risultati del comando in un editor di testo normale ed eliminare quanto segue:

   - Eliminare `{ "deploymentGroupInfo":` all'inizio dell'output.
   - Eliminare `}` alla fine dell'output.
   - Eliminare la riga contenente `deploymentGroupId`.
   - Eliminare la riga contenente `deploymentGroupName`.

   Il contenuto del file di testo dovrebbe essere analogo al seguente:

```
{
    "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. Modificare i parametri, se necessario. Per informazioni sui parametri di configurazione dei trigger, vedere. TriggerConfig
4. Salvare gli aggiornamenti come file JSON, quindi eseguire il comando update-deployment-group tramite l'opzione `--cli-input-json`. Assicurati di includere l'`--current-deployment-group-name`opzione e di sostituire il nome del tuo file JSON con: *`filename`*

> **Important**
>
> Assicurarsi di includere `file://` prima del nome del file. Questo è obbligatorio in questo comando.

```
aws deploy update-deployment-group --current-deployment-group-name deployment-group-name --cli-input-json file://filename.json
```

Al termine del processo di creazione riceverai un messaggio di notifica di prova che indica che i dettagli delle autorizzazioni e dei trigger sono impostati correttamente.

Eliminare un trigger da un gruppo di distribuzione CodeDeploy

Dal momento che esiste un limite di 10 trigger per gruppo di distribuzione, è possibile eliminare i trigger non più utilizzati. Non è possibile annullare l'eliminazione di un trigger, ma è possibile ricreare un trigger eliminato.

Eliminare un trigger da un gruppo di distribuzione (console)

1. Accedi a Console di gestione AWS e apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Accedi con lo stesso utente che hai configuratoGuida introduttiva con CodeDeploy.

2. Nel riquadro di navigazione, espandi Distribuisci, quindi scegli Applicazioni.
3. Nella pagina Applications (Applicazioni) scegliere il nome dell'applicazione associata al gruppo di distribuzione in cui si intende eliminare un trigger.

4. Nella pagina Application details (Dettagli applicazione) scegliere il gruppo di distribuzione in cui si desidera eliminare un trigger.
5. Scegli Modifica.
6. Espandere Advanced - optional (Opzioni avanzate - opzionale).
7. Nell'area Triggers (Trigger) scegliere il trigger che si desidera eliminare, quindi scegliere Delete trigger (Elimina trigger).
8. Scegli Save changes (Salva modifiche).

Eliminare un trigger da un gruppo di distribuzione (CLI)

Per utilizzare la CLI per eliminare un trigger, chiama il comando update-deployment-group, con parametri di configurazione del trigger vuoti, specificando:

- Il nome dell'applicazione associata a questo gruppo di distribuzione. Per visualizzare un elenco di nomi di applicazioni, chiamate il comando list-applications.
- Il nome del gruppo di distribuzione associato a questa applicazione. Per visualizzare un elenco di nomi di gruppi di distribuzione, chiamate il comando list-deployment-groups.

Ad esempio:

```
aws deploy update-deployment-group --application-name application-name --current-
deployment-group-name deployment-group-name --trigger-configurations
```

Formati CodeDeploy di dati JSON per i trigger

Puoi utilizzare l'output JSON creato quando un trigger per una distribuzione o un'istanza viene attivato in un flusso di lavoro di notifica personalizzato, ad esempio l'invio di messaggi alle code di Amazon SQS o l'invocazione di una funzione in. AWS Lambda

> **Note**
>
> In questa guida non viene spiegato come configurare le notifiche tramite JSON. Per informazioni sull'utilizzo di Amazon SNS per inviare messaggi alle code di Amazon SQS, consulta Invio di messaggi Amazon SNS alle code Amazon SQS. Per informazioni sull'uso di Amazon SNS per richiamare una funzione Lambda, consulta Richiamare le funzioni Lambda utilizzando le notifiche Amazon SNS.

I seguenti esempi illustrano la struttura dell'output JSON disponibile con i trigger CodeDeploy.

Instance-Based Esempio di output JSON per Triggers

```
{
    "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
    ]
}
```

Esempio di output JSON per Trigger Deployment-Based

```
{
    "region": "us-west-1",
    "accountId": "111222333444",
    "eventTriggerName": "Trigger-group-us-west-3-deploy-failed",
    "applicationName": "ProductionApp-us-west-3",
    "deploymentId": "d-75I7MBT7C",
    "deploymentGroupName": "dep-group-def-456",
    "createTime": "1446744188.595",
    "completeTime": "1446744190.402",
    "deploymentOverview": {
        "Failed": "10",
        "InProgress": "0",
```

```
        "Pending": "0",
        "Skipped": "0",
        "Succeeded": "0"
    },
    "status": "Failed",
    "errorInformation": {
        "ErrorCode": "IAM_ROLE_MISSING",
        "ErrorMessage": "IAM Role is missing for deployment group: dep-group-def-456"
    }
}
```

Sicurezza in AWS CodeDeploy

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. Third-party i revisori testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei programmi di AWS conformità. Per ulteriori informazioni sui programmi di conformità applicabili AWS CodeDeploy, consulta AWS Services in Scope by Compliance Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo CodeDeploy. I seguenti argomenti mostrano come eseguire la configurazione CodeDeploy per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere CodeDeploy le tue risorse.

Argomenti

- Protezione dei dati in AWS CodeDeploy
- Gestione delle identità e degli accessi per AWS CodeDeploy
- Registrazione e monitoraggio CodeDeploy
- Convalida della conformità per AWS CodeDeploy
- Resilienza in AWS CodeDeploy
- Sicurezza dell'infrastruttura in AWS CodeDeploy

Protezione dei dati in AWS CodeDeploy

Ulteriori informazioni su come il modello di responsabilità condivisa AWS si applica alla protezione dei dati in AWS CodeDeploy. Come descritto in questo modello, AWS è responsabile della protezione

dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, consulta Domande frequenti sulla privacy dei dati . Per ulteriori informazioni sulla protezione dei dati in Europa, consulta il Centro generale sulla protezione dei dati (GDPR).

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e configurare singoli utenti con Centro identità AWS IAM o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori o Servizi AWS utilizzi la console, l'API CodeDeploy o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Riservatezza del traffico Internet

CodeDeploy è un servizio di distribuzione completamente gestito che supporta istanze EC2, funzioni Lambda, Amazon ECS e server locali. Per le istanze EC2 e i server locali, un agente basato su host comunica tramite TLS. CodeDeploy

Attualmente, la comunicazione dall'agente al servizio richiede una connessione Internet in uscita in modo che l'agente possa comunicare con gli endpoint del servizio pubblico CodeDeploy e Amazon S3. In un virtual private cloud, questo può essere realizzato con un gateway Internet, una connessione VPN da sito a sito alla rete aziendale o una connessione diretta.

L' CodeDeploy agente supporta i proxy HTTP.

Gli endpoint Amazon VPC, con tecnologia AWS PrivateLink, sono disponibili CodeDeploy in alcune regioni. Per informazioni dettagliate, vedi Utilizzo CodeDeploy con Amazon Virtual Private Cloud.

> **Note**
>
> L' CodeDeploy agente è necessario solo se esegui la distribuzione su una piattaforma di EC2/On-premises elaborazione Amazon. L'agente non è necessario per le distribuzioni che utilizzano Amazon ECS o AWS Lambda la piattaforma di calcolo.

Crittografia dei dati a riposo

Il codice cliente non è memorizzato in. CodeDeploy Come servizio di distribuzione, CodeDeploy consiste nell'invio di comandi all' CodeDeploy agente in esecuzione su istanze EC2 o server locali. L' CodeDeploy agente esegue quindi i comandi utilizzando TLS. I dati del modello di servizio per le distribuzioni, la configurazione di distribuzione, i gruppi di distribuzione, le applicazioni e le revisioni delle applicazioni vengono archiviati in Amazon DynamoDB e crittografati su disco utilizzando un file di proprietà e gestito da. Chiave di proprietà di AWS CodeDeploy Per ulteriori informazioni, consulta s.Chiave di proprietà di AWS

Crittografia dei dati in transito

L' CodeDeploy agente avvia tutte le comunicazioni CodeDeploy tramite la porta 443. L'agente esegue il polling CodeDeploy e resta in ascolto di un comando. L' CodeDeploy agente è open source. Tutte le comunicazioni da servizio a servizio e da client a servizio sono crittografate in transito utilizzando TLS. Ciò protegge i dati dei clienti in transito tra CodeDeploy e altri servizi come Amazon S3.

Gestione delle chiavi crittografiche

Non ci sono chiavi di crittografia da gestire. I dati del modello di CodeDeploy servizio vengono crittografati utilizzando un Chiave di proprietà di AWS file di proprietà e gestito da CodeDeploy. Per ulteriori informazioni, vedere Chiave di proprietà di AWS s.

Gestione delle identità e degli accessi per AWS CodeDeploy

AWS Identity and Access Management (IAM) aiuta un Servizio AWS amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. CodeDeploy IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso tramite policy
- In che modo AWS CodeDeploy funziona con IAM
- AWS politiche gestite (predefinite) per CodeDeploy
- CodeDeploy aggiornamenti a AWS policy gestite
- AWS CodeDeploy esempi di politiche basate sull'identità
- Risoluzione dei problemi AWS CodeDeploy identità e accesso
- CodeDeploy riferimento alle autorizzazioni
- Cross-service confusa prevenzione sostitutiva

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta Risoluzione dei problemi AWS CodeDeploy identità e accesso)
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta In che modo AWS CodeDeploy funziona con IAM)
- Amministratore IAM: scrivi policy per gestire l'accesso (consulta AWS CodeDeploy esempi di politiche basate sull'identità)

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come Centro identità AWS IAM (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/ Facebook Per ulteriori informazioni sull'accesso, consulta Come accedere all' Account AWS nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta AWS Signature Version 4 per le richieste API nella Guida per l'utente di IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali come utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'utente di IAM.

Utenti e gruppi

Un utente IAM è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee nella Guida per l'utente IAM.

Un gruppo IAM specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente di IAM.

Ruoli IAM

Un ruolo IAM è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo passando da un ruolo utente a un ruolo IAM (console) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta Metodi per assumere un ruolo nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Identity-based politiche

Identity-based le politiche sono documenti di policy sulle autorizzazioni JSON che alleghi a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Identity-based le politiche possono essere politiche in linea (incorporate direttamente in una singola identità) o politiche gestite (politiche autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scegliere tra policy gestite e policy in linea nella Guida per l'utente di IAM.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse da tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Policy di controllo dei servizi (SCP): specifica il numero massimo di autorizzazioni per un'organizzazione o un'unità organizzativa (OU) in AWS Organizations. Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Policy di controllo delle risorse (RCP): imposta le autorizzazioni massime disponibili per le risorse degli account. Per ulteriori informazioni, consulta Policy di controllo delle risorse (RCP) nella Guida per l'utente di AWS Organizations .
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

In che modo AWS CodeDeploy funziona con IAM

Prima di utilizzare IAM per gestire l'accesso a CodeDeploy, è necessario comprendere con quali funzionalità IAM è disponibile l'uso CodeDeploy. Per ulteriori informazioni, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Argomenti

- CodeDeploy politiche basate sull'identità
- CodeDeploy politiche basate sulle risorse
- CodeDeploy Autorizzazione basata sui tag
- CodeDeploy ruoli IAM

CodeDeploy politiche basate sull'identità

Con le policy basate sull'identità IAM, puoi specificare azioni e risorse consentite o negate e le condizioni in base alle quali le azioni sono consentite o negate. CodeDeploy supporta azioni, risorse

e chiavi di condizione. Per informazioni sugli elementi utilizzati in una policy JSON, consulta il riferimento agli elementi della policy JSON di IAM nella IAM User Guide.

Azioni

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni di eseguire l'operazione associata.

Le azioni politiche CodeDeploy utilizzano il `codedeploy:` prefisso che precede l'azione. Ad esempio, l'autorizzazione `codedeploy:GetApplication` fornisce all'utente le autorizzazioni per eseguire l'operazione `GetApplication`. Le dichiarazioni politiche devono includere un `NotAction` elemento `Action` or. CodeDeploy definisce il proprio set di azioni che descrivono le attività che è possibile eseguire con questo servizio.

Per specificare più azioni in una sola istruzione, separa ciascuna di esse con una virgola come mostrato di seguito:

```
"Action": [
      "codedeploy:action1",
      "codedeploy:action2"
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, includi la seguente operazione per specificare tutte le operazioni che iniziano con la parola `Describe`:

```
"Action": "ec2:Describe*"
```

Per un elenco di CodeDeploy azioni, consulta Actions Defined by AWS CodeDeploy nella IAM User Guide.

Per una tabella che elenca tutte le azioni CodeDeploy API e le risorse a cui si applicano, consultaCodeDeploy riferimento alle autorizzazioni.

Resources

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Ad esempio, puoi indicare un gruppo di distribuzione (*myDeploymentGroup*) nella tua dichiarazione utilizzando il relativo ARN come segue:

```
"Resource": "arn:aws:codedeploy:us-
west-2:123456789012:deploymentgroup:myApplication/myDeploymentGroup"
```

È inoltre possibile specificare tutti i gruppi di distribuzione che appartengono a un account utilizzando il carattere jolly (*) come segue:

```
"Resource": "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:*"
```

Per specificare tutte le risorse o se un'operazione API non supporta gli ARN, utilizza il carattere jolly (*) nell'elemento `Resource` come segue:

```
"Resource": "*"
```

Alcune azioni CodeDeploy API accettano più risorse (ad esempio,`BatchGetDeploymentGroups`). Per specificare più risorse in una sola dichiarazione, separa i relativi ARN con una virgola come mostrato di seguito:

```
"Resource": ["arn1", "arn2"]
```

CodeDeploy fornisce una serie di operazioni per utilizzare le CodeDeploy risorse. Per un elenco di operazioni disponibili, consulta la sezione CodeDeploy riferimento alle autorizzazioni.

Per un elenco dei tipi di CodeDeploy risorse e dei relativi ARN, consulta Resources Defined by AWS CodeDeploy nella IAM User Guide. Per informazioni sulle azioni in cui è possibile specificare l'ARN di ogni risorsa, vedere Azioni definite da. AWS CodeDeploy

CodeDeploy risorse e operazioni

In CodeDeploy, la risorsa principale è un gruppo di distribuzione. In una policy, utilizzi un Amazon Resource Name (ARN) per identificare la risorsa a cui si applica la policy. CodeDeploy supporta altre risorse che possono essere utilizzate con gruppi di distribuzione, tra cui applicazioni, configurazioni di distribuzione e istanze. e vi si può fare riferimento come risorse secondarie. Queste risorse e risorse secondarie hanno ARN univoci associati. Per ulteriori informazioni, consulta Amazon Resource Names (ARN) nel Riferimenti generali di Amazon Web Services.

Tipo di risorsa	Formato ARN
Gruppo di distribuzione	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:deploymentgroup:` *`application-name`* `/`*`deployment-group-name`*
Applicazione	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:application:` *`application-name`*
Configurazione dell'implementazione	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:deploymentconfig:` *`deployment-configuration-name`*
Istanza	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:instance/` *`instance-ID`*
Tutte le risorse CodeDeploy	`arn:aws:codedeploy:*`
Tutte CodeDeploy le risorse di proprietà dell'account specificato nella regione specificata	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:*`

> **Note**
>
> La maggior parte dei servizi in uso AWS considera i due punti (:) o una barra (/) come lo stesso carattere negli ARN. Tuttavia, CodeDeploy utilizza una corrispondenza esatta nei modelli e nelle regole delle risorse. Assicurati di utilizzare i caratteri ARN corretti durante la

creazione di modelli di eventi, facendo in modo che corrispondano alla sintassi ARN nella risorsa.

Chiavi di condizione

CodeDeploy non fornisce chiavi di condizione specifiche del servizio, ma supporta l'uso di alcune chiavi di condizione globali. Per ulteriori informazioni, consultare Chiavi di contesto delle condizioni globali AWS nella Guida per l'utente IAM.

Esempi

Per visualizzare esempi di politiche basate sull' CodeDeploy identità, vedere. AWS CodeDeploy esempi di politiche basate sull'identità

CodeDeploy politiche basate sulle risorse

CodeDeploy non supporta politiche basate sulle risorse. Per visualizzare un esempio di una pagina dettagliata dei criteri basati sulle risorse, vedere Utilizzo delle politiche basate sulle risorse per. AWS Lambda

CodeDeploy Autorizzazione basata sui tag

CodeDeploy non supporta l'etichettatura delle risorse o il controllo dell'accesso in base ai tag.

CodeDeploy ruoli IAM

Un ruolo IAM è un'entità nel tuo AWS account che dispone di autorizzazioni specifiche.

Utilizzo di credenziali temporanee con CodeDeploy

È possibile utilizzare credenziali temporanee per effettuare l'accesso con la federazione, assumere un ruolo IAM o un ruolo multi-account. È possibile ottenere credenziali di sicurezza temporanee chiamando operazioni AWS STS API come AssumeRoleo. GetFederationToken

CodeDeploy supporta l'uso di credenziali temporanee.

Service-linked ruoli

CodeDeploy non supporta i ruoli collegati ai servizi.

Ruoli dei servizi

Questa funzionalità consente a un servizio di assumere un ruolo di servizio per conto dell'utente. Questo ruolo consente al servizio di accedere alle risorse in altri servizi per completare un'azione per conto dell'utente. I ruoli di servizio vengono visualizzati nell' AWS account e sono di proprietà dell'account. Ciò significa che un utente può modificare le autorizzazioni per questo ruolo. Tuttavia, il farlo potrebbe pregiudicare la funzionalità del servizio.

CodeDeploy supporta i ruoli di servizio.

Scelta di un ruolo IAM in CodeDeploy

Quando crei una risorsa del gruppo di distribuzione in CodeDeploy, devi scegliere un ruolo per consentire l'accesso CodeDeploy ad Amazon EC2 per tuo conto. Se hai creato in precedenza un ruolo del servizio o un ruolo collegato ai servizi, CodeDeploy fornisce un elenco di ruoli tra cui scegliere. È importante scegliere un ruolo che consenta l'accesso per avviare e arrestare le istanze EC2.

AWS politiche gestite (predefinite) per CodeDeploy

AWS affronta molti casi d'uso comuni fornendo policy IAM autonome create e amministrate da AWS. Queste politiche AWS gestite concedono le autorizzazioni per i casi d'uso più comuni, in modo da evitare di dover verificare quali autorizzazioni sono necessarie. Per ulteriori informazioni, consultare Policy gestite da AWS nella Guida per l'utente di IAM.

Argomenti

- Elenco di AWS politiche gestite per CodeDeploy
- CodeDeploy politiche e notifiche gestite

Elenco di AWS politiche gestite per CodeDeploy

Le seguenti politiche AWS gestite, che puoi allegare agli utenti del tuo account, sono specifiche per CodeDeploy:

- `AWSCodeDeployFullAccess`: concede accesso completo a CodeDeploy.

> **Note**
>
> AWSCodeDeployFullAccess non fornisce autorizzazioni per le operazioni in altri servizi necessari per distribuire le tue applicazioni, come Amazon EC2 e Amazon S3, solo per operazioni specifiche di. CodeDeploy

- `AWSCodeDeployDeployerAccess`: concede l'autorizzazione a registrare e distribuire le revisioni.

- `AWSCodeDeployReadOnlyAccess`: concede accesso in sola lettura a CodeDeploy.

- `AWSCodeDeployRole`: Consente di: CodeDeploy
  - leggi i tag sulle tue istanze o identifica le tue istanze Amazon EC2 tramite i nomi dei gruppi di Amazon EC2 Auto Scaling
  - leggi, crea, aggiorna ed elimina gruppi, lifecycle hook, policy di scalabilità e funzionalità di warm pool di Amazon EC2 Auto Scaling
  - pubblicare informazioni su argomenti di Amazon SNS
  - recuperare informazioni sugli allarmi Amazon CloudWatch
  - leggi e aggiorna le risorse nel servizio Elastic Load Balancing

  La policy contiene il codice seguente:

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction",
        "autoscaling:DeleteLifecycleHook",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeLifecycleHooks",
        "autoscaling:PutLifecycleHook",
        "autoscaling:RecordLifecycleActionHeartbeat",
        "autoscaling:CreateAutoScalingGroup",
```

```
        "autoscaling:CreateOrUpdateTags",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling:EnableMetricsCollection",
        "autoscaling:DescribePolicies",
        "autoscaling:DescribeScheduledActions",
        "autoscaling:DescribeNotificationConfigurations",
        "autoscaling:SuspendProcesses",
        "autoscaling:ResumeProcesses",
        "autoscaling:AttachLoadBalancers",
        "autoscaling:AttachLoadBalancerTargetGroups",
        "autoscaling:PutScalingPolicy",
        "autoscaling:PutScheduledUpdateGroupAction",
        "autoscaling:PutNotificationConfiguration",
        "autoscaling:DescribeScalingActivities",
        "autoscaling:DeleteAutoScalingGroup",
        "autoscaling:PutWarmPool",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:TerminateInstances",
        "tag:GetResources",
        "sns:Publish",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutMetricAlarm",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeLoadBalancerAttributes",
        "elasticloadbalancing:DescribeInstanceHealth",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
        "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetGroupAttributes",
        "elasticloadbalancing:DescribeTargetHealth",
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:DeregisterTargets"
      ],
      "Resource": "*"
    }
  ]
}
```

- `AWSCodeDeployRoleForLambda`: concede CodeDeploy l'autorizzazione all'accesso AWS Lambda e a qualsiasi altra risorsa richiesta per una distribuzione.

- `AWSCodeDeployRoleForECS`: concede l' CodeDeploy autorizzazione ad accedere ad Amazon ECS e a qualsiasi altra risorsa richiesta per una distribuzione.

- `AWSCodeDeployRoleForECSLimited`: concede l' CodeDeployautorizzazione ad accedere ad Amazon ECS e a qualsiasi altra risorsa richiesta per una distribuzione con le seguenti eccezioni:
  - Nella `hooks` sezione del AppSpec file, è `CodeDeployHook_` possibile utilizzare solo le funzioni Lambda con nomi che iniziano con. Per ulteriori informazioni, consulta AppSpec sezione 'hook' per una distribuzione Amazon ECS.
  - L'accesso al bucket S3 è limitato ai bucket S3 con un tag di registrazione, `UseWithCodeDeploy`, che ha un valore di `true`. Per ulteriori informazioni, consulta Etichettatura degli oggetti.
- `AmazonEC2RoleforAWSCodeDeployLimited`: concede CodeDeploy l'autorizzazione per ottenere ed elencare oggetti in un bucket CodeDeploy Amazon S3. La policy contiene il seguente codice:

  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"
                }
            }
        }
    ]
}
```

Le autorizzazioni per alcuni aspetti del processo di distribuzione sono concesse ad altri due tipi di ruolo che agiscono per conto di CodeDeploy:

- Un profilo di istanza IAM è un ruolo IAM che colleghi alle tue istanze Amazon EC2. Questo profilo include le autorizzazioni necessarie per accedere ai bucket GitHub o ai repository Amazon S3 in cui sono archiviate le applicazioni. Per ulteriori informazioni, consulta Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2.
- Un ruolo di servizio è un ruolo IAM che concede le autorizzazioni a un AWS servizio in modo che possa accedere alle risorse. AWS Le policy associate al ruolo di servizio determinano a quali AWS risorse il servizio può accedere e le azioni che può eseguire con tali risorse. Infatti CodeDeploy, un ruolo di servizio viene utilizzato per quanto segue:
  - Per leggere i tag applicati alle istanze o i nomi dei gruppi Amazon EC2 Auto Scaling associati alle istanze. Ciò consente di CodeDeploy identificare le istanze in cui può distribuire applicazioni.
  - Per eseguire operazioni su istanze, gruppi Amazon EC2 Auto Scaling e sistemi di bilanciamento del carico Elastic Load Balancing.
  - Pubblicare informazioni su argomenti di Amazon SNS in modo da poter inviare notifiche quando si verificano eventi di distribuzione o istanza specifici.
  - Per recuperare informazioni sugli CloudWatch allarmi per configurare il monitoraggio degli allarmi per le distribuzioni.

  Per ulteriori informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.

Puoi anche creare policy IAM personalizzate per concedere autorizzazioni per azioni e risorse. CodeDeploy Alleghi queste policy personalizzate ai ruoli IAM e poi assegni i ruoli agli utenti o ai gruppi che richiedono le autorizzazioni.

CodeDeploy politiche e notifiche gestite

CodeDeploy supporta le notifiche per informare gli utenti di importanti modifiche alle implementazioni. Le politiche gestite CodeDeploy includono dichiarazioni politiche per la funzionalità di notifica. Per ulteriori informazioni, vedere Cosa sono le notifiche?.

Autorizzazioni per le notifiche nelle politiche gestite ad accesso completo

Le policy gestite `AWSCodeDeployFullAccess` includono le seguenti dichiarazioni per consentire l'accesso completo alle notifiche. Gli utenti a cui viene applicata questa policy gestita possono anche creare e gestire argomenti di Amazon SNS per le notifiche, iscrivere e annullare l'iscrizione degli utenti agli argomenti ed elencare argomenti da scegliere come obiettivi per le regole di notifica.

```
    {
        "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": "*"
    }
```

Autorizzazioni per le notifiche nelle politiche gestite di sola lettura

Le policy gestite `AWSCodeDeployReadOnlyAccess` includono le seguenti dichiarazioni per consentire l'accesso in sola lettura alle notifiche. Gli utenti con questa policy gestita applicata possono visualizzare le notifiche per le risorse, ma non possono crearle, gestirle o sottoscriverle.

```
   {
        "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": "*"
    }
```

Autorizzazioni per le notifiche in altre politiche gestite

La politica `AWSCodeDeployDeployerAccess` gestita include le seguenti istruzioni per consentire agli utenti di creare, aggiornare, sottoscrivere e visualizzare le notifiche relative alle risorse, ma non possono eliminarle. Gli utenti a cui viene applicata questa policy gestita possono anche creare e gestire argomenti Amazon SNS per le notifiche.

Questa policy include le autorizzazioni per eseguire le seguenti operazioni:

- `codestar-notifications:CreateNotificationRule`— Consente ai mandanti di creare notifiche.
- `codestar-notifications:DescribeNotificationRule`— Consente ai responsabili di recuperare informazioni sulle notifiche.
- `codestar-notifications:UpdateNotificationRule`— Consente ai mandanti di aggiornare le notifiche.
- `codestar-notifications:Subscribe`— Consente ai mandanti di sottoscrivere gli aggiornamenti delle notifiche.
- `codestar-notifications:Unsubscribe`— Consente ai responsabili di annullare l'iscrizione agli aggiornamenti delle notifiche.
- `codestar-notifications:ListNotificationRules`— Consente ai responsabili di recuperare l'elenco delle regole di notifica.
- `codestar-notifications:ListTargets`— Consente ai responsabili di recuperare l'elenco degli obiettivi.
- `codestar-notifications:ListTagsforResource`— Consente ai mandanti di recuperare l'elenco dei tag.
- `codestar-notifications:ListEventTypes`— Consente ai responsabili di recuperare l'elenco dei tipi di eventi.
- `chatbot:DescribeSlackChannelConfiguration`— Consente ai responsabili di recuperare informazioni sulle configurazioni dei canali Slack.

- `sns:ListTopics`— Consente ai responsabili di recuperare l'elenco degli argomenti di Amazon SNS per le notifiche.

```
  {
     "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" : "*"
    }
```

Per ulteriori informazioni, consulta Gestione delle identità e degli accessi per AWS CodeStar Notifications.

CodeDeploy aggiornamenti a AWS policy gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite CodeDeploy da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per gli avvisi automatici sulle modifiche apportate a questa pagina, iscriversi al feed RSS alla pagina CodeDeploy Cronologia dei documenti.

Modifica	Descrizione	Data
`AWSCodeDeployDeployerAccess` politica gestita: aggiornamenti alla politica esistente	È stata aggiornata l'`codestar-notifications:NotificationsForResource` azione per supportare le modifiche di convalida delle policy IAM. La risorsa originale `arn:aws:codedeploy:*` è stata aggiornata a. `arn:aws:codedeploy:*:*:application:*` Per ulteriori informazioni su questa policy, consulta Autorizzazioni per le notifiche in altre politiche gestite.	16 dicembre 2024
`AWSCodeDeployReadOnlyAccess` politica gestita: aggiornamenti alla politica esistente	È stata aggiornata l'`codestar-notifications:NotificationsForResource` azione per	16 dicembre 2024

Modifica	Descrizione	Data
	supportare le modifiche di convalida delle policy IAM. La risorsa originale `arn:aws:codedeploy:*` è stata aggiornata a. `arn:aws:codedeploy:*:*:application:*` Per ulteriori informazioni su questa policy, consulta Autorizzazioni per le notifiche nelle politiche gestite di sola lettura.	
`AWSCodeDeployFullAccess` politica gestita: aggiornamenti alla politica esistente	È stata aggiornata l'`codestar-notifications:NotificationsForResource` azione per supportare le modifiche di convalida delle policy IAM. La risorsa originale `arn:aws:codedeploy:*` è stata aggiornata a. `arn:aws:codedeploy:*:*:application:*` Per ulteriori informazioni su questa policy, consulta Autorizzazioni per le notifiche nelle politiche gestite ad accesso completo.	16 dicembre 2024

Modifica	Descrizione	Data
`AWSCodeDeployRole` politica gestita: aggiornamenti alla politica esistente	Sono state aggiunte le `elasticloadbalancing:DescribeTargetGroupAttributes` azioni `elasticloadbalancing:DescribeLoadBalancerAttributes` e all'informativa sulla politica per supportare le modifiche di Elastic Load Balancing. Per ulteriori informazioni su questa policy, consulta AWSCodeDeployRole.	16 agosto 2023
`AWSCodeDeployFullAccess` policy gestita: aggiornamenti alla policy esistente	È stata aggiunta l'`chatbot:ListMicrosoftTeamsChannelConfigurations` azione alla dichiarazione politica per supportare le modifiche alle notifiche. Per ulteriori informazioni su questa policy, consulta AWSCodeDeployRole.	11 maggio 2023

Modifica	Descrizione	Data
`AWSCodeDeployRole` politica gestita: aggiornamenti alla politica esistente	È stata aggiunta l'`autoscaling:CreateOrUpdateTags` azione alla dichiarazione politica per supportare le modifiche all'autorizzazione di Amazon EC2 Auto Scaling. Per ulteriori informazioni su questa policy, consulta AWSCodeDeployRole.	3 febbraio 2023
`AmazonEC2RoleforAWSCodeDeployLimited` politica gestita: aggiornamenti alla politica esistente	È stata rimossa l'`s3:ListBucket` azione dalla dichiarazione politica che include la `s3:ExistingObjectTag/UseWithCodeDeploy` condizione. Per ulteriori informazioni su questa policy, consulta AmazonEC2RoleforAWSCodeDeployLimited.	22 novembre 2021
`AWSCodeDeployRole` politica gestita: aggiornamenti alla politica esistente	È stata aggiunta l'`autoscaling:PutWarmPool` azione per supportare l'aggiunta di pool caldi ai gruppi Amazon EC2 Auto Scaling blue/green per le distribuzioni. Sono state rimosse le azioni duplicate inutili.	18 maggio 2021

Modifica	Descrizione	Data
CodeDeploy ha iniziato a tenere traccia delle modifiche	CodeDeploy ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	18 maggio 2021

AWS CodeDeploy esempi di politiche basate sull'identità

Per impostazione predefinita, gli utenti non sono autorizzati a creare o modificare CodeDeploy risorse. Inoltre, non possono eseguire attività utilizzando l' AWS API Console di gestione AWS AWS CLI, o. È necessario creare policy IAM che concedano ai ruoli IAM l'autorizzazione a eseguire operazioni API sulle risorse specificate di cui hanno bisogno. È quindi necessario collegare tali ruoli IAM a utenti o gruppi che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consultare Creazione di policy nella scheda JSON nella Guida per l'utente IAM.

Nel CodeDeploy, le politiche basate sull'identità vengono utilizzate per gestire le autorizzazioni alle varie risorse relative al processo di distribuzione. È possibile controllare l'accesso a tutti i seguenti tipi di risorse:

- Applicazioni e revisioni dell'applicazione.
- Distribuzioni.
- Configurazioni delle distribuzioni.
- Istanze e istanze locali.

Le funzionalità controllate dalle policy delle risorse variano a seconda del tipo di risorsa, come illustrato nella seguente tabella:

Tipi di risorse	Funzionalità
Tutti	Aprire ed elencare i dettagli sulle risorse
Applicazioni	Creare risorse

Tipi di risorse	Funzionalità
Configurazioni della distribuzione Gruppi di distribuzione	Delete resources (Elimina risorse)
Distribuzioni	Creare distribuzione Arrestare distribuzione
Revisioni dell'applicazione	Registrare le revisioni dell'applicazione
Applicazioni Gruppi di distribuzione	Aggiornare le risorse
On-premises istanze	Aggiungere tag alle istanze Rimuovere tag dalle istanze Registrare istanze Annullare registrazione delle istanze

Il seguente esempio mostra una policy di autorizzazioni che consente a un utente di eliminare il gruppo di distribuzione denominato **`WordPress_DepGroup`** associato con l'applicazione denominata **`WordPress_App`** nella regione **`us-west-2`**.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:DeleteDeploymentGroup"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:deploymentgroup:WordPress_App/
WordPress_DepGroup"
```

```
      ]
    }
  ]
}
```

Argomenti

- Customer-managed esempi di politiche
- Best practice per le policy
- Utilizzo della console CodeDeploy
- Consentire agli utenti di visualizzare le loro autorizzazioni

Customer-managed esempi di politiche

In questa sezione, puoi trovare esempi di politiche che concedono autorizzazioni per varie azioni. CodeDeploy Queste politiche funzionano quando utilizzi l' CodeDeploy API, gli AWS SDK o il. AWS CLIÈ necessario concedere autorizzazioni aggiuntive per le operazioni eseguite nella console. Per ulteriori informazioni sulla concessione delle autorizzazioni della console, consultare Utilizzo della console CodeDeploy.

> **Note**
>
> Tutti gli esempi utilizzano la regione Stati Uniti occidentali (Oregon) (`us-west-2`) e contengono ID account fittizi.

Esempi

- Esempio 1: consenti l'autorizzazione a eseguire CodeDeploy operazioni in una singola regione
- Esempio 2: consentire l'autorizzazione a registrare le revisioni per una singola applicazione
- Esempio 3: consentire l'autorizzazione a creare distribuzioni per un singolo gruppo di distribuzione

Esempio 1: consenti l'autorizzazione a eseguire CodeDeploy operazioni in una singola regione

L'esempio seguente concede le autorizzazioni per eseguire CodeDeploy operazioni solo nella **`us-west-2`** regione:

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:*"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:*"
      ]
    }
  ]
}
```

Esempio 2: consentire l'autorizzazione a registrare le revisioni per una singola applicazione

Nell'esempio seguente vengono concesse autorizzazioni per registrare le revisioni dell'applicazione per tutte le applicazioni che iniziano con **`Test`** nella regione **`us-west-2`**:

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:RegisterApplicationRevision"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:application:Test*"
      ]
    }
  ]
}
```

Esempio 3: consentire l'autorizzazione a creare distribuzioni per un singolo gruppo di distribuzione

L'esempio seguente consente l'autorizzazione a creare distribuzioni per il gruppo di distribuzione denominato **WordPress_DepGroup** associato all'applicazione denominata**WordPress_App**, la configurazione di distribuzione personalizzata denominata e tutte le revisioni dell'applicazione associate all'applicazione **ThreeQuartersHealthy** denominata. **WordPress_App** Tutte queste risorse sono nella regione **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 practice per le policy

Identity-based le politiche determinano se qualcuno può creare, accedere o eliminare CodeDeploy risorse nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consultare la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e suggerimenti utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente di IAM.

- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare Protezione dell'accesso API con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console CodeDeploy

Se utilizzi la CodeDeploy console, devi disporre di un set minimo di autorizzazioni che ti consenta di descrivere altre AWS risorse per il tuo AWS account. Per CodeDeploy utilizzarla nella CodeDeploy console, è necessario disporre delle autorizzazioni dei seguenti servizi:

- Amazon EC2 Auto Scaling
- AWS CodeDeploy
- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- AWS Identity and Access Management
- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon CloudWatch

Se crei una policy IAM più restrittiva delle autorizzazioni minime richieste, la console non funzionerà come previsto per gli utenti che hanno un ruolo in quella policy IAM. Per garantire che tali utenti possano continuare a utilizzare la CodeDeploy console, collega anche la policy `AWSCodeDeployReadOnlyAccess` gestita al ruolo assegnato all'utente, come descritto in. AWS politiche gestite (predefinite) per CodeDeploy

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l' CodeDeploy API.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica

include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Risoluzione dei problemi AWS CodeDeploy identità e accesso

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con un CodeDeploy IAM.

Argomenti

- Non sono autorizzato a eseguire iam: PassRole
- Voglio consentire l'accesso a persone esterne al mio AWS account per accedere alle mie CodeDeploy risorse

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, le tue policy devono essere aggiornate per poter passare un ruolo a CodeDeploy.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in CodeDeploy. Tuttavia, l'operazione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per trasmettere il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire l'accesso a persone esterne al mio AWS account per accedere alle mie CodeDeploy risorse

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo degli accessi (ACL), utilizzare tali policy per concedere alle persone l'accesso alle proprie risorse.

Per maggiori informazioni, consulta gli argomenti seguenti:

- Per sapere se CodeDeploy supporta queste funzionalità, consultaIn che modo AWS CodeDeploy funziona con IAM.

- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente di IAM.

CodeDeploy riferimento alle autorizzazioni

Utilizza la tabella seguente quando configuri politiche di accesso e scrittura delle autorizzazioni che puoi allegare a un'identità IAM (politiche basate sull'identità). La tabella elenca ogni operazione CodeDeploy API, le azioni per le quali è possibile concedere le autorizzazioni per eseguire l'azione e il formato della risorsa ARN da utilizzare per concedere le autorizzazioni. Puoi specificare le operazioni nel campo `Action` della policy. Puoi specificare un ARN, con o senza un carattere jolly (*), come valore della risorsa nel campo `Resource` della policy.

Puoi utilizzare le chiavi AWS-wide condition nelle tue CodeDeploy politiche per esprimere condizioni. Per un elenco completo delle chiavi AWS-wide, consulta Available keys nella IAM User Guide.

Per specificare un'operazione, utilizza il prefisso `codedeploy:` seguito dal nome dell'operazione API (ad esempio, `codedeploy:GetApplication` and `codedeploy:CreateApplication`). Per specificare più operazioni in una sola istruzione, separa ciascuna di esse con una virgola (ad esempio, `"Action": ["codedeploy:action1", "codedeploy:action2"]`).

Utilizzo di caratteri jolly

Puoi usare un carattere jolly (*) nell'ARN per specificare più operazioni o risorse. Ad esempio, `codedeploy:*` specifica tutte le CodeDeploy azioni e `codedeploy:Get*` specifica tutte le CodeDeploy azioni che iniziano con la parola. `Get` L'esempio seguente consente di concedere l'accesso a tutti i gruppi di distribuzione con nomi che iniziano con `West` e sono associati alle applicazioni che hanno nomi iniziano con `Test`.

```
arn:aws:codedeploy:us-west-2:444455556666:deploymentgroup:Test*/West*
```

Puoi usare i caratteri jolly solo con le seguenti risorse elencate nella tabella:

- *application-name*
- *deployment-group-name*
- *deployment-configuration-name*
- *instance-ID*

I caratteri jolly non possono essere usati con o. *region* *account-id* Per ulteriori informazioni sui caratteri jolly, consulta Identificatori IAM nella Guida per l'utente IAM.

> **Note**
>
> Nell'ARN per ciascuna operazione la risorsa è seguita da due punti (:). La risorsa può anche essere seguita da una barra (/). Per ulteriori informazioni, vedi CodeDeployesempi di ARN.

CodeDeploy Operazioni API e autorizzazioni richieste per le azioni

AddTagsToOnPremisesInstances

Operazione: `codedeploy:AddTagsToOnPremisesInstances`

Richieste per aggiungere tag a una o più istanze locali.

Risorsa: `arn:aws:codedeploy:`*region*`:`*account-id*`:instance/`*instance-ID*

BatchGetApplicationRevisions

Operazione: `codedeploy:BatchGetApplicationRevisions`

Richieste per ottenere informazioni sulle diverse revisioni dell'applicazione associate all'utente .

Risorsa: `arn:aws:codedeploy:`*region*`:`*account-id*`:application:`*application-name*

BatchGetApplications

Operazione: `codedeploy:BatchGetApplications`

Richieste per ottenere informazioni sulle diverse applicazioni associate all'utente .

Risorsa: `arn:aws:codedeploy:`*region*`:`*account-id*`:application:*`

BatchGetDeploymentGroups

Operazione: `codedeploy:BatchGetDeploymentGroups`

Richieste per ottenere informazioni sui diversi gruppi di distribuzione associate all'utente .

Risorsa: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

BatchGetDeploymentInstances

Operazione: `codedeploy:BatchGetDeploymentInstances`

Richieste per ottenere informazioni su una o più istanze che fanno parte di un gruppo di distribuzione.

Risorsa: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

BatchGetDeployments

Operazione: `codedeploy:BatchGetDeployments`

Richieste per ottenere informazioni sulle diverse distribuzioni associate all'utente .

Risorsa: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

BatchGetOnPremisesInstances

Operazione: `codedeploy:BatchGetOnPremisesInstances`

Richieste per ottenere informazioni su una o più istanze locali.

Risorsa: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:*`

ContinueDeployment

Operazione: `codedeploy:ContinueDeployment`

Necessario durante una blue/green distribuzione per iniziare a registrare le istanze in un ambiente sostitutivo con un sistema di bilanciamento del carico Elastic Load Balancing.

Risorsa: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

CreateApplication

Operazione: `codedeploy:CreateApplication`

Richieste per creare un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

CreateDeployment[1]

Operazione: `codedeploy:CreateDeployment`

Richieste per la creazione di una distribuzione per un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

CreateDeploymentConfig

Operazione: `codedeploy:CreateDeploymentConfig`

Richieste per la creazione di una configurazione della distribuzione personalizzata associata all'utente .

Risorsa: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentconfig/`*`deployment-configuration-name`*

CreateDeploymentGroup

Operazione: `codedeploy:CreateDeploymentGroup`

Richieste per la creazione di un gruppo di distribuzione per un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

DeleteApplication

Operazione: `codedeploy:DeleteApplication`

Richieste per eliminare un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

DeleteDeploymentConfig

Operazione: `codedeploy:DeleteDeploymentConfig`

Richieste per eliminare una configurazione della distribuzione personalizzata associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentconfig/deployment-configuration-name`

DeleteDeploymentGroup

Operazione: `codedeploy:DeleteDeploymentGroup`

Richieste per eliminare un gruppo di distribuzione per un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

DeregisterOnPremisesInstance

Operazione: `codedeploy:DeregisterOnPremisesInstance`

Richieste per l'annullamento della registrazione di un'istanza locale.

Risorsa: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

GetApplication

Operazione: `codedeploy:GetApplication`

Richieste per ottenere informazioni su una singola applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:application:application-name`

GetApplicationRevision

Operazione: `codedeploy:GetApplicationRevision`

Richieste per ottenere informazioni su una singola revisione dell'applicazione per un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:application:application-name`

GetDeployment

Operazione: `codedeploy:GetDeployment`

Richieste per ottenere informazioni su una singola distribuzione a un gruppo di distribuzione associato all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

GetDeploymentConfig

Operazione: `codedeploy:GetDeploymentConfig`

Richieste per ottenere informazioni su una singola configurazione di distribuzione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentconfig/deployment-configuration-name`

GetDeploymentGroup

Operazione: `codedeploy:GetDeploymentGroup`

Richieste per ottenere informazioni su un singolo gruppo di distribuzione per un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

GetDeploymentInstance

Operazione: `codedeploy:GetDeploymentInstance`

Richieste per ottenere informazioni su una singola istanza in una distribuzione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

GetOnPremisesInstance

Operazione: `codedeploy:GetOnPremisesInstance`

Richieste per ottenere informazioni su una singola istanza locale.

Risorsa: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

ListApplicationRevisions

Operazione: `codedeploy:ListApplicationRevisions`

Richieste per ottenere informazioni su tutte le revisioni dell'applicazione per un'applicazione associate all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:application:*`

ListApplications

Operazione: `codedeploy:ListApplications`

Richieste per ottenere informazioni su tutte le applicazioni associate all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:application:*`

ListDeploymentConfigs

Operazione: `codedeploy:ListDeploymentConfigs`

Richieste per ottenere informazioni su tutte le configurazioni di distribuzione associate all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentconfig/*`

ListDeploymentGroups

Operazione: `codedeploy:ListDeploymentGroups`

Richieste per ottenere informazioni su tutti i gruppi di distribuzione per un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/*`

ListDeploymentInstances

Operazione: `codedeploy:ListDeploymentInstances`

Necessario per ottenere informazioni su tutte le istanze di una distribuzione associate all'utente.

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

ListDeployments

Operazione: `codedeploy:ListDeployments`

Necessario per fornire informazioni su tutte le distribuzioni a un gruppo di distribuzione associato all'utente o per associare tutte le distribuzioni all'utente.

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

ListGitHubAccountTokenNames

Operazione: `codedeploy:ListGitHubAccountTokenNames`

Necessario per ottenere un elenco dei nomi delle connessioni archiviate agli account. GitHub

Risorsa: `arn:aws:codedeploy:region:account-id:*`

ListOnPremisesInstances

Operazione: `codedeploy:ListOnPremisesInstances`

Richieste per ottenere un elenco di uno o più nomi di istanza locali.

Risorsa: `arn:aws:codedeploy:region:account-id:*`

RegisterApplicationRevision

Operazione: `codedeploy:RegisterApplicationRevision`

Richieste per registrare informazioni su una revisione dell'applicazione per un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:application:application-name`

RegisterOnPremisesInstance

Operazione: `codedeploy:RegisterOnPremisesInstance`

Necessario per registrare un'istanza locale con CodeDeploy.

Risorsa: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

RemoveTagsFromOnPremisesInstances

Operazione: `codedeploy:RemoveTagsFromOnPremisesInstances`

Richieste per rimuovere tag da una o più istanze locali.

Risorsa: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

SkipWaitTimeForInstanceTermination

Operazione: `codedeploy:SkipWaitTimeForInstanceTermination`

Necessario per ignorare un tempo di attesa specificato e iniziare a terminare le istanze nell'ambiente originale subito dopo che il traffico è stato instradato correttamente in una distribuzione. blue/green

Risorsa: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

StopDeployment

Operazione: `codedeploy:StopDeployment`

Richieste per arrestare una distribuzione in corso a un gruppo di distribuzione per un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

UpdateApplication[3]

Operazione: `codedeploy:UpdateApplication`

Richieste per modificare le informazioni su un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:application:application-name`

UpdateDeploymentGroup[3]

Operazione: `codedeploy:UpdateDeploymentGroup`

Richieste per modificare le informazioni su un singolo gruppo di distribuzione per un'applicazione associata all'utente .

Risorsa: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

[1] Quando si specificano `CreateDeployment` le autorizzazioni, è necessario specificare anche `GetDeploymentConfig` le autorizzazioni per la configurazione di distribuzione `GetApplicationRevision` e/o le `RegisterApplicationRevision` autorizzazioni per la revisione dell'applicazione.

[2] Valido per `ListDeployments` quando si fornisce un gruppo di distribuzione, ma non quando si elencano tutte le distribuzioni associate all'utente.

[3] Infatti`UpdateApplication`, è necessario disporre `UpdateApplication` delle autorizzazioni sia per il vecchio che per il nuovo nome dell'applicazione. Per le operazioni `UpdateDeploymentGroup`

che comportano la modifica del nome di un gruppo di distribuzione, è necessario disporre delle autorizzazioni `UpdateDeploymentGroup` per il nome del gruppo di distribuzione vecchio e nuovo.

Cross-service confusa prevenzione sostitutiva

Il problema confused deputy è un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire un'azione può costringere un'entità maggiormente privilegiata a eseguire l'azione. Nel AWS, l'impersonificazione intersettoriale può portare al confuso problema del vicesceriffo. Cross-service l'impersonificazione può verificarsi quando un servizio (il servizio chiamante) chiama un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare che ciò accada, AWS mette a disposizione strumenti che consentono di proteggere i dati relativi a tutti i servizi con responsabili del servizio a cui è stato concesso l'accesso alle risorse dell'account.

Ti consigliamo di utilizzare le chiavi aws: SourceArn e aws: SourceAccount global condition context nelle politiche delle risorse per limitare le autorizzazioni che CodeDeploy forniscono un altro servizio alla risorsa. Se si utilizzano entrambe le chiavi di contesto delle condizioni globali e il valore `aws:SourceArn` contiene l'ID account, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account nella stessa dichiarazione di policy. Utilizzare `aws:SourceArn` se si desidera consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizza `aws:SourceAccount` se desideri che qualsiasi risorsa in quell'account sia associata all'uso tra servizi.

Per EC2/On-Premises le distribuzioni AWS Lambda e Amazon ECS regolari, il valore di `aws:SourceArn` dovrebbe includere il CodeDeploy gruppo di distribuzione ARN con cui CodeDeploy è consentito assumere il ruolo IAM.

Per le blue/greenimplementazioni Amazon ECS create tramite CloudFormation, il valore di `aws:SourceArn` dovrebbe includere lo stack CloudFormation ARN con cui CodeDeploy è consentito assumere il ruolo IAM.

Il modo più efficace per proteggersi dal confuso problema del vice è utilizzare la `aws:SourceArn` chiave con l'ARN completo della risorsa. Se non conosci l'ARN completo o se stai specificando più risorse, usa caratteri jolly (*) per le parti sconosciute.

Ad esempio, puoi utilizzare la seguente policy di fiducia con una distribuzione EC2/On-Premises Amazon ECS AWS Lambda o normale:

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/*"
                }
            }
        }
    ]
}
```

Per una blue/greendistribuzione Amazon ECS creata tramite CloudFormation, puoi utilizzare:

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/*"
                }
            }
        }
    ]
}
```

Registrazione e monitoraggio CodeDeploy

In questa sezione viene fornita una panoramica del monitoraggio, della registrazione e della risposta agli eventi imprevisti CodeDeploy.

Controllo di tutte le interazioni con CodeDeploy

CodeDeploy è integrato con AWS CloudTrail, un servizio che acquisisce le chiamate API effettuate da o per conto del CodeDeploy tuo AWS account e invia i file di registro a un bucket S3 da te specificato. CloudTrail acquisisce le chiamate API dalla CodeDeploy console, dai CodeDeploy comandi tramite o direttamente AWS CLI dalle API. CodeDeploy Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare a quale richiesta è stata effettuata CodeDeploy, l'indirizzo IP di origine da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e così via. Per ulteriori informazioni CloudTrail, consulta Lavorare con i file di CloudTrail registro nella Guida AWS CloudTrail per l'utente.

Puoi visualizzare i dati di log creati da una CodeDeploy distribuzione configurando l' CloudWatchagente Amazon per visualizzare i dati aggregati nella CloudWatch console o accedendo a un'istanza per esaminare il file di registro. Per ulteriori informazioni, consulta Invia i log degli CodeDeploy agenti a CloudWatch.

Gestione degli avvisi e degli incidenti

Puoi utilizzare Amazon CloudWatch Events per rilevare e reagire ai cambiamenti nello stato di un'istanza o di una distribuzione (un evento) nelle tue CodeDeploy operazioni. Quindi, in base alle regole che crei, CloudWatch Events richiama una o più azioni mirate quando una distribuzione o un'istanza entra nello stato specificato in una regola. A seconda del tipo di cambiamento di stato,

potrebbe essere necessario inviare notifiche, acquisire informazioni sullo stato, eseguire un'azione correttiva, avviare eventi o eseguire altre operazioni. Puoi selezionare i seguenti tipi di obiettivi quando utilizzi CloudWatch Events come parte delle tue CodeDeploy operazioni:

- AWS Lambda funzioni
- Flussi Kinesis
- Code Amazon SQS SQS
- Built-in obiettivi (azioni di allarme) CloudWatch
- Argomenti di Amazon SNS

Di seguito sono riportati alcuni casi d'uso:

- Utilizza una funzione di Lambda per inoltrare una notifica a un canale Slack quando le distribuzioni hanno esito negativo.
- Effettua il push dei dati relativi a distribuzioni o istanze su un flusso Kinesis per supportare il monitoraggio completo dello stato in tempo reale.
- Utilizza le azioni di CloudWatch allarme per arrestare, terminare, riavviare o ripristinare automaticamente le istanze EC2 quando si verifica un evento di distribuzione o istanza specificato.

Per ulteriori informazioni, consulta What is Amazon CloudWatch Events nella Amazon CloudWatch User Guide.

Convalida della conformità per AWS CodeDeploy

La nostra nuova esperienza AWS di registrazione non è progettata per carichi di lavoro regolamentati. Se utilizzi la nostra nuova esperienza di AWS registrazione, ma desideri utilizzarla AWS per carichi di lavoro regolamentati, puoi iscriverti a AWS (avanzata) o attivare funzionalità avanzate per il tuo ambiente. AWS

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Ambito per programma di conformitàServizi AWS e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta la Documentazione AWS sulla sicurezza.

Resilienza in AWS CodeDeploy

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta infrastruttura globale.AWS

Sicurezza dell'infrastruttura in AWS CodeDeploy

In quanto servizio gestito, AWS CodeDeploy è protetto dalle procedure di sicurezza della rete AWS globale descritte nel white paper Amazon Web Services: panoramica dei processi di sicurezza.

Utilizzi chiamate API AWS pubblicate per accedere CodeDeploy attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.2 o versioni successive. È consigliabile TLS 1.3 o versioni successive. I client devono inoltre supportare suite di crittografia con Perfect Forward Secrecy (PFS), come Ephemeral (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta che è associata a un'entità IAM. In alternativa, è possibile utilizzare Servizio di token di sicurezza AWS (AWS STS) per generare le credenziali di sicurezza temporanee per sottoscrivere le richieste.

Riferimento

Riferimento.

Argomenti

- CodeDeploy AppSpec riferimento al file
- CodeDeploy riferimento alla configurazione dell'agente
- CloudFormation modelli di CodeDeploy riferimento
- Utilizzo CodeDeploy con Amazon Virtual Private Cloud
- CodeDeploy riferimento al kit di risorse
- CodeDeploy quote

CodeDeploy AppSpec riferimento al file

Questa sezione è solo per riferimento. Per una panoramica concettuale del AppSpec file, vedereApplication Specification Files.

Il file delle specifiche dell'applicazione (AppSpec file) è un file in formato YAML o utilizzato da CodeDeploy per JSON-formatted gestire una distribuzione.

> **Note**
>
> Il AppSpec file per una EC2/On-Premises distribuzione deve avere un nome`appspec.yml`, a meno che non si stia eseguendo una distribuzione locale. Per ulteriori informazioni, consulta Crea una distribuzione locale.

Argomenti

- AppSpec file su una piattaforma di elaborazione Amazon ECS
- AppSpec file su un AWS Lambda piattaforma di calcolo
- AppSpec file su una piattaforma di EC2/on-premises elaborazione
- AppSpec Struttura dei file
- AppSpec Esempio di file
- AppSpec Spaziatura tra i file

- Convalida il AppSpec file e la posizione del file

AppSpec file su una piattaforma di elaborazione Amazon ECS

Per le applicazioni della piattaforma di calcolo Amazon ECS, il AppSpec file viene utilizzato CodeDeploy per determinare:

- Il tuo file di definizione delle attività di Amazon ECS. Questo è specificato con il relativo ARN nelle `TaskDefinition` istruzioni del file. AppSpec
- Il contenitore e la porta del set di attività sostitutivo in cui l'Application Load Balancer o il Network Load Balancer reindirizzano il traffico durante una distribuzione. Questo viene specificato con le istruzioni contenute nel file. `LoadBalancerInfo` AppSpec
- Informazioni opzionali sul tuo servizio Amazon ECS, come la versione della piattaforma su cui viene eseguito, le sue sottoreti e i suoi gruppi di sicurezza.
- Funzioni Lambda opzionali da eseguire durante gli hook che corrispondono agli eventi del ciclo di vita durante una distribuzione Amazon ECS. Per ulteriori informazioni, consulta AppSpec sezione 'hook' per una distribuzione Amazon ECS.

AppSpec file su un AWS Lambda piattaforma di calcolo

Per le applicazioni della piattaforma di calcolo AWS Lambda, il AppSpec file viene utilizzato per CodeDeploy determinare:

- Quale versione della funzione Lambda distribuire.
- Quali funzioni Lambda utilizzare come test di convalida.

Un AppSpec file può essere YAML-formatted o. JSON-formatted È inoltre possibile inserire il contenuto di un AppSpec file direttamente nella CodeDeploy console quando si crea una distribuzione.

AppSpec file su una piattaforma di EC2/on-premises elaborazione

Se l'applicazione utilizza la piattaforma di EC2/On-Premises elaborazione, il AppSpec file deve avere un YAML-formatted nome `appspec.yml` e deve essere inserito nella radice della struttura di directory del codice sorgente dell'applicazione. In alternativa, le distribuzioni non riescono. Viene utilizzato per CodeDeploy determinare:

- Cosa deve installare sulle istanze a partire dalla revisione dell'applicazione in Amazon S3 o. GitHub
- Quali hook di eventi del ciclo di vita eseguire in risposta agli eventi del ciclo di vita della distribuzione.

Una volta completato, AppSpec il file viene raggruppato, insieme al contenuto da distribuire, in un file di archivio (zip, tar o tar compresso). Per ulteriori informazioni, consulta Utilizzo delle revisioni delle applicazioni per CodeDeploy.

> **Note**
>
> I formati di file di archivio tar e tar compressi (.tar e .tar.gz) non sono supportati per le istanze di Windows Server.

Dopo aver creato un file di archivio in bundle (noto CodeDeploy come revisione), lo carichi in un bucket Amazon S3 o in un repository Git. Quindi lo usi per distribuire la revisione CodeDeploy . Per istruzioni, consulta l'argomento Crea una distribuzione con CodeDeploy.

L'appspec.yml per l'implementazione di una piattaforma di EC2/On-Premises calcolo viene salvato nella directory principale della revisione. Per ulteriori informazioni, consulta Aggiungi un AppSpec file per una distribuzione EC2/On-Premises e Pianifica una revisione per CodeDeploy.

AppSpec Struttura dei file

Di seguito è riportata la struttura di alto livello di un AppSpec file utilizzato per le distribuzioni su piattaforme AWS Lambda e di calcolo. EC2/On-Premises

Un valore in un YAML-formatted AppSpec file che è una stringa non deve essere racchiuso tra virgolette («») se non diversamente specificato.

AppSpec struttura dei file per le distribuzioni di Amazon ECS

> **Note**
>
> Questo AppSpec file è scritto in YAML, ma puoi usare la stessa struttura per scriverne uno in JSON. Una stringa in un JSON-formatted AppSpec file è sempre racchiusa tra virgolette («»).

```
version: 0.0
resources:
  ecs-service-specifications
hooks:
  deployment-lifecycle-event-mappings
```

In questa struttura:

versione

Questa sezione specifica la versione del file. AppSpec Non modificare questo valore. È obbligatoria. Attualmente, l'unico valore consentito è **`0.0`**. È riservato CodeDeploy per usi futuri.

Specifica version con una stringa.

resources

Questa sezione specifica le informazioni sull'applicazione Amazon ECS da distribuire.

Per ulteriori informazioni, consulta AppSpec sezione «risorse» per le distribuzioni di Amazon ECS.

hooks

Questa sezione specifica le funzioni Lambda da eseguire in specifici hook di eventi del ciclo di vita della distribuzione per convalidare la distribuzione.

Per ulteriori informazioni, consulta Elenco degli hook relativi agli eventi del ciclo di vita per una distribuzione Amazon ECS.

AppSpec struttura dei file per AWS Implementazioni Lambda

> **Note**
>
> Questo AppSpec file è scritto in YAML, ma puoi usare la stessa struttura per scrivere un AppSpec file per una distribuzione Lambda in JSON. Una stringa in un JSON-formatted AppSpec file è sempre racchiusa tra virgolette («»).

```
version: 0.0
resources:
  lambda-function-specifications
```

```
hooks:
  deployment-lifecycle-event-mappings
```

In questa struttura:

versione

Questa sezione specifica la versione del file. AppSpec Non modificare questo valore. È obbligatoria. Attualmente, l'unico valore consentito è **`0.0`**. È riservato CodeDeploy per usi futuri.

Specifica version con una stringa.

resources

Questa sezione specifica le informazioni sulla funzione Lambda da distribuire.

Per ulteriori informazioni, consulta AppSpec sezione «risorse» (Amazon ECS e AWS Lambda (solo distribuzioni).

hooks

Questa sezione specifica le funzioni Lambda da eseguire in occasione di eventi specifici del ciclo di vita della distribuzione per convalidare la distribuzione.

Per ulteriori informazioni, consulta AppSpec sezione 'hooks'.

AppSpec struttura EC2/On-Premises dei file per le distribuzioni

```
version: 0.0
os: operating-system-name
files:
  source-destination-files-mappings
permissions:
  permissions-specifications
hooks:
  deployment-lifecycle-event-mappings
```

In questa struttura:

versione

Questa sezione specifica la versione del file. AppSpec Non modificare questo valore. È obbligatoria. Attualmente, l'unico valore consentito è **`0.0`**. È riservato CodeDeploy per usi futuri.

Specifica version con una stringa.

os

In questa sezione viene specificato il valore del sistema operativo dell'istanza in cui effettui la distribuzione. È obbligatoria. È possibile specificare i seguenti valori:

- linux: l'istanza è un'istanza Amazon Linux, Ubuntu Server, RHEL o macOS.
- windows: l'istanza è un'istanza di Windows Server.

Specifica os con una stringa.

files

In questa sezione vengono specificati i nomi dei file che devono essere copiati nell'istanza durante l'evento Install della distribuzione.

Per ulteriori informazioni, consulta AppSpec sezione 'file' (solo distribuzioni) EC2/On-Premises.

permissions

In questa sezione viene specificato come le autorizzazioni speciali, se presenti, devono essere applicate alla sezione `files` quando vengono copiate nell'istanza. Questa sezione si applica solo alle istanze Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) e macOS.

Per ulteriori informazioni, consultare AppSpec sezione 'permessi' (solo distribuzioni) EC2/On-Premises.

hooks

In questa sezione vengono specificati gli script da eseguire per specifici eventi del ciclo di vita durante la distribuzione.

Per ulteriori informazioni, consulta AppSpec sezione 'hooks'.

Argomenti

- AppSpec sezione 'file' (solo distribuzioni) EC2/On-Premises
- AppSpec sezione «risorse» (Amazon ECS e AWS Lambda (solo distribuzioni)
- AppSpec sezione 'permessi' (solo distribuzioni) EC2/On-Premises
- AppSpec sezione 'hooks'

AppSpec sezione 'file' (solo distribuzioni) EC2/On-Premises

Fornisce informazioni CodeDeploy sui file della revisione dell'applicazione da installare sull'istanza durante l'evento di installazione della distribuzione. Questa sezione è obbligatoria solo se stai copiando i file dalla tua revisione nelle posizioni sull'istanza durante la distribuzione.

Questa sezione ha la seguente struttura:

```
files:
  - source: source-file-location-1
    destination: destination-file-location-1
file_exists_behavior: DISALLOW|OVERWRITE|RETAIN
```

È possibile impostare più coppie `source` e `destination`.

L'istruzione `source` identifica un file o una directory della revisione da copiare nell'istanza:

- Se `source` fa riferimento a un file, solo i file specificati vengono copiati nell'istanza.
- Se `source` fa riferimento a una directory, tutti i file della directory vengono copiati nell'istanza.
- Se `source` è una barra singola («/» per le istanze di Amazon Linux, RHEL e Ubuntu Server o «\» per le istanze di Windows Server), tutti i file della revisione vengono copiati nell'istanza.

I percorsi utilizzati in `source` sono relativi al `appspec.yml` file, che dovrebbe essere alla radice della revisione. Per i dettagli sulla struttura dei file di una revisione, vedere. Pianifica una revisione per CodeDeploy

L'istruzione `destination` identifica il percorso dell'istanza in cui i file devono essere copiati. Deve trattarsi di un percorso completo, ad esempio `/root/destination/directory` (su Linux, RHEL e Ubuntu) o `c:\destination\folder` (su Windows).

`source` e `destination` sono specificati ciascuno con una stringa.

L'`file_exists_behavior`istruzione è facoltativa e specifica in che modo CodeDeploy gestisce i file già esistenti in una posizione di destinazione di distribuzione ma che non facevano parte della precedente distribuzione riuscita. Questa impostazione può assumere uno qualsiasi dei seguenti valori:

- DISALLOW: la distribuzione non riesce. Questo è anche il comportamento predefinito se non viene specificata alcuna opzione.

- SOVRASCRITTURA: la versione del file contenuta nella revisione dell'applicazione attualmente distribuita sostituisce la versione già presente sull'istanza.
- RETAIN: la versione del file già presente nell'istanza viene conservata e utilizzata come parte della nuova distribuzione.

Quando usi l'`file_exists_behavior`impostazione, tieni presente che questa impostazione:

- può essere specificata una sola volta e si applica a tutti i file e le directory elencati sotto`files:`.
- ha la precedenza sull'`--file-exists-behavior` AWS CLI opzione e sull'opzione `fileExistsBehavior` API (entrambe anch'esse opzionali).

Ecco una `files` sezione di esempio per un'istanza Amazon Linux, Ubuntu Server o RHEL.

```
files:
  - source: Config/config.txt
    destination: /webapps/Config
  - source: source
    destination: /webapps/myApp
```

In questo esempio, le seguenti due operazioni vengono eseguite durante l'evento Install:

1. Copia il file `Config/config.txt` della revisione nel percorso `/webapps/Config/config.txt` dell'istanza.
2. Copia in modo ricorsivo tutti i file della directory `source` della revisione nella directory `/webapps/myApp` dell'istanza.

Esempi della sezione 'File'

I seguenti esempi mostrano come specificare la sezione `files`. Sebbene questi esempi descrivano le strutture di file e directory (cartelle) di Windows Server, possono essere facilmente adattati per Amazon Linux, Ubuntu Server e istanze RHEL.

> **Note**
>
> Solo le EC2/On-Premises distribuzioni utilizzano la sezione. `files` Non si applica alle distribuzioni AWS Lambda.

Per i seguenti esempi presupponiamo che questi file siano nel bundle della directory radice di `source`:

- `appspec.yml`
- `my-file.txt`
- `my-file-2.txt`
- `my-file-3.txt`

```
# 1) Copy only my-file.txt to the destination folder c:\temp.
#
files:
  - source: .\my-file.txt
    destination: c:\temp
#
# Result:
#   c:\temp\my-file.txt
#
# ---------------------
#
# 2) Copy only my-file-2.txt and my-file-3.txt to the destination folder c:\temp.
#
files:
  - source: my-file-2.txt
    destination: c:\temp
  - source: my-file-3.txt
    destination: c:\temp
#
# Result:
#   c:\temp\my-file-2.txt
#   c:\temp\my-file-3.txt
#
# ---------------------
#
# 3) Copy my-file.txt, my-file-2.txt, and my-file-3.txt (along with the appspec.yml
 file) to the destination folder c:\temp.
#
files:
  - source: \
    destination: c:\temp
#
# Result:
```

```
#   c:\temp\appspec.yml
#   c:\temp\my-file.txt
#   c:\temp\my-file-2.txt
#   c:\temp\my-file-3.txt
```

Per i seguenti esempi presupponiamo che il file `appspec.yml` sia nel bundle della radice di `source` insieme a una cartella denominata `my-folder` che contiene tre file:

- `appspec.yml`
- `my-folder\my-file.txt`
- `my-folder\my-file-2.txt`
- `my-folder\my-file-3.txt`

```
# 4) Copy the 3 files in my-folder (but do not copy my-folder itself) to the
 destination folder c:\temp.
#
files:
  - source: .\my-folder
    destination: c:\temp
#
# Result:
#   c:\temp\my-file.txt
#   c:\temp\my-file-2.txt
#   c:\temp\my-file-3.txt
#
# ---------------------
#
# 5) Copy my-folder and its 3 files to my-folder within the destination folder c:\temp.
#
files:
  - source: .\my-folder
    destination: c:\temp\my-folder
#
# Result:
#   c:\temp\my-folder\my-file.txt
#   c:\temp\my-folder\my-file-2.txt
#   c:\temp\my-folder\my-file-3.txt
#
# ---------------------
#
```

```
# 6) Copy the 3 files in my-folder to other-folder within the destination folder c:
\temp.
#
files:
  - source: .\my-folder
    destination: c:\temp\other-folder
#
# Result:
#   c:\temp\other-folder\my-file.txt
#   c:\temp\other-folder\my-file-2.txt
#   c:\temp\other-folder\my-file-3.txt
#
# ----------------------
#
# 7) Copy only my-file-2.txt and my-file-3.txt to my-folder within the destination
 folder c:\temp.
#
files:
  - source: .\my-folder\my-file-2.txt
    destination: c:\temp\my-folder
  - source: .\my-folder\my-file-3.txt
    destination: c:\temp\my-folder
#
# Result:
#   c:\temp\my-folder\my-file-2.txt
#   c:\temp\my-folder\my-file-3.txt
#
# ----------------------
#
# 8) Copy only my-file-2.txt and my-file-3.txt to other-folder within the destination
 folder c:\temp.
#
files:
  - source: .\my-folder\my-file-2.txt
    destination: c:\temp\other-folder
  - source: .\my-folder\my-file-3.txt
    destination: c:\temp\other-folder
#
# Result:
#   c:\temp\other-folder\my-file-2.txt
#   c:\temp\other-folder\my-file-3.txt
#
# ----------------------
#
```

```
# 9) Copy my-folder and its 3 files (along with the appspec.yml file) to the
 destination folder c:\temp. If any of the files already exist on the instance,
 overwrite them.
#
files:
  - source: \
    destination: c:\temp
file_exists_behavior: OVERWRITE
#
# Result:
#   c:\temp\appspec.yml
#   c:\temp\my-folder\my-file.txt
#   c:\temp\my-folder\my-file-2.txt
#   c:\temp\my-folder\my-file-3.txt
```

AppSpec sezione «risorse» (Amazon ECS e AWS Lambda (solo distribuzioni)

Il contenuto nella `'resources'` sezione del AppSpec file varia a seconda della piattaforma di elaborazione della distribuzione. La `'resources'` sezione relativa a una distribuzione di Amazon ECS contiene la definizione delle attività di Amazon ECS, il contenitore e la porta per instradare il traffico verso il set di attività Amazon ECS aggiornato e altre informazioni opzionali. La `'resources'` sezione relativa a una AWS Lambda distribuzione contiene il nome, l'alias, la versione corrente e la versione di destinazione di una funzione Lambda.

Argomenti

- AppSpec sezione 'risorse' per AWS Implementazioni Lambda
- AppSpec sezione «risorse» per le distribuzioni di Amazon ECS

AppSpec sezione 'risorse' per AWS Implementazioni Lambda

La `'resources'` sezione specifica la funzione Lambda da distribuire e ha la seguente struttura:

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"
            }
        }
    }
]
```

Ogni proprietà viene specificata con una stringa.

- `name`: obbligatorio Questo è il nome della funzione Lambda da distribuire.
- `alias`: obbligatorio Questo è il nome dell'alias della funzione Lambda.
- `currentversion`: obbligatorio Questa è la versione della funzione Lambda a cui punta attualmente il traffico. Questo valore deve essere un numero intero positivo valido.
- `targetversion`: obbligatorio Questa è la versione della funzione Lambda a cui viene spostato il traffico. Questo valore deve essere un numero intero positivo valido.

AppSpec sezione «risorse» per le distribuzioni di Amazon ECS

La `'resources'` sezione specifica il servizio Amazon ECS da distribuire e ha la seguente struttura:

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
                    }
                ]
            }
        }
    }
]
```

Ogni proprietà è specificata con una stringa ad eccezione di`ContainerPort`, che è un numero.

- `TaskDefinition`: obbligatorio Questa è la definizione dell'attività da distribuire per il servizio Amazon ECS. Viene specificata con l'ARN della definizione dell'attività. Il formato dell'ARN è `arn:aws:ecs:`*`aws-region`*`:`*`account-id`*`:task-definition/`*`task-definition-family`*`:`*`task-definition-revision`*. Per ulteriori informazioni, consulta Amazon Resource Names (ARN) e AWS service namespace.

  > **Note**
  >
  > La `:`*`task-definition-revision`* parte dell'ARN è facoltativa. Se viene omesso, Amazon ECS utilizza l'ultima revisione ACTIVE della definizione dell'attività.

- `ContainerName`: obbligatorio Questo è il nome del contenitore Amazon ECS che contiene la tua applicazione Amazon ECS. Deve essere un contenitore specificato nella definizione dell'attività di Amazon ECS.
- `ContainerPort`: obbligatorio Questa è la porta del container verso cui verrà indirizzato il traffico.
- `PlatformVersion`: facoltativo. La versione della piattaforma delle attività Fargate nel servizio Amazon ECS distribuito. Per ulteriori informazioni, consulta Versioni della piattaforma AWS Fargate. Se non specificato, `LATEST` viene utilizzato per impostazione predefinita.

- `NetworkConfiguration`: facoltativo. In `AwsvpcConfiguration`, è possibile specificare le forme seguenti: Per ulteriori informazioni, consulta AwsVpcConfigurationAmazon ECS Container Service API Reference.
  - `Subnets`: facoltativo. Un elenco separato da virgole di una o più sottoreti nel tuo servizio Amazon ECS.
  - `SecurityGroups`: facoltativo. Un elenco separato da virgole di uno o più gruppi di sicurezza nel tuo Amazon Elastic Container Service.
  - `AssignPublicIp`: facoltativo. Una stringa che specifica se l'interfaccia di rete elastica del servizio Amazon ECS riceve un indirizzo IP pubblico. I valori validi sono `ENABLED` e `DISABLED`.

  > **Note**
  >
  > È necessario specificare tutte o nessuna delle impostazioni in `NetworkConfiguration`. Ad esempio, se si desidera specificare `Subnets`, è necessario specificare anche `SecurityGroups` e `AssignPublicIp`. Se non ne viene specificata nessuna, CodeDeploy utilizza le impostazioni di rete correnti di Amazon ECS.

- `CapacityProviderStrategy`: facoltativo. Un elenco di provider di capacità Amazon ECS che desideri utilizzare per la distribuzione. Per ulteriori informazioni, consulta i provider di capacità di Amazon ECS nella Amazon Elastic Container Service Developer Guide. Per ogni provider di capacità, puoi specificare le seguenti impostazioni. Per i dettagli su queste impostazioni, consulta AWS: :ECS:: ServiceCapacityProviderStrategyItem nella Guida per l'utente AWS CloudFormation
  - `Base`: facoltativo. Il valore di base indica il numero minimo di attività da eseguire nel provider di capacità specificato. Solo un provider di capacità in una strategia di provider di capacità può avere una base definita. Se non viene specificato alcun valore, sarà utilizzato il valore predefinito 0.
  - `CapacityProvider`: facoltativo. Nome breve o completo del provider di capacità. Esempio: CapacityProvidera
  - `Weight`: facoltativo.

    Il valore peso indica la percentuale relativa del numero totale di attività avviate che devono utilizzare il provider di capacità specificato. Il valore `weight` viene preso in considerazione dopo che è stato soddisfatto il valore `base`, se definito.

    Se non viene specificato alcun valore `weight`, sarà utilizzato il valore predefinito `0`. Quando più provider di capacità sono specificati nell'ambito di una strategia di provider di capacità,

almeno uno dei provider deve avere un valore di peso superiore a zero e qualsiasi provider di capacità con un peso pari a `0` non sarà utilizzato per svolgere attività. Se specifichi più provider di capacità in una strategia tutti con un peso di `0`, qualsiasi operazione `RunTask` o `CreateService` che utilizza la strategia del provider di capacità avrà esito negativo.

Uno scenario di esempio di utilizzo dei pesi è la definizione di una strategia contenente due provider di capacità, ognuno con un peso pari a `1`, quindi una volta soddisfatto il `base`, le attività saranno suddivise in maniera equa tra i due provider. Utilizzando la stessa logica, se si specifica un peso di 1 per capacityProviderA e un peso di `4` per capacityProviderB, per ogni attività che viene eseguita utilizzando capacityProviderA, quattro attività utilizzano capacityProviderB.

AppSpec sezione 'permessi' (solo distribuzioni) EC2/On-Premises

La `'permissions'` sezione specifica in che modo le eventuali autorizzazioni speciali devono essere applicate ai file e directories/folders nella `'files'` sezione dopo che sono state copiate nell'istanza. Puoi specificare più istruzioni `object`. Questa sezione è facoltativa. Si applica solo alle istanze Amazon Linux, Ubuntu Server, RHEL e macOS.

> **Note**
>
> Nelle istanze macOS, le `context` istruzioni `acls` e non sono supportate. Una distribuzione il cui AppSpec file utilizza uno di essi su macOS fallisce prima dell'installazione dei file. Le altre istruzioni funzionano come descritto in questa sezione.

> **Note**
>
> La sezione `'permissions'` viene utilizzata solo per le distribuzioni EC2/On-Premises. Non viene utilizzato per le implementazioni AWS Lambda o Amazon ECS.

Questa sezione ha la seguente struttura:

```
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
```

Sono disponibili le seguenti istruzioni:

- `object`: obbligatorio Si tratta di un insieme di oggetti del file system (file o directories/folders) a cui vengono applicate le autorizzazioni specificate dopo che gli oggetti del file system sono stati copiati nell'istanza.

  Specificare `object` con una stringa.

- `pattern`: opzionale. Specifica un modello per applicare le autorizzazioni. Se non è specificato o è specificato con i caratteri speciali **"**"**, le autorizzazioni vengono applicate a tutti i file o le directory corrispondenti, a seconda del `type`.

  Specifica `pattern` con una stringa tra virgolette ("").

- `except`: opzionale. Specifica i file o le directory che sono eccezioni di `pattern`.

  Specifica `except` con un elenco di stringhe separate da virgola tra parentesi quadre.

- `owner`: opzionale. Il nome del proprietario di `object`. Se non specificato, tutti i proprietari esistenti applicati al file o alla directory/folder struttura originale rimangono invariati dopo l'operazione di copia.

  Specificare `owner` con una stringa.

- `group`: opzionale. Il nome del gruppo per `object`. Se non specificato, tutti i gruppi esistenti applicati al file o alla directory/folder struttura originale rimangono invariati dopo l'operazione di copia.

  Specificare `group` con una stringa.

- `mode`: opzionale. Un valore numerico che specifica le autorizzazioni a cui applicare. `object` L'impostazione della modalità segue la sintassi del comando Linux chmod.

> **Important**
>
> Se il valore include uno zero iniziale, è necessario racchiuderlo tra virgolette doppie o rimuovere lo zero iniziale in modo che rimangano solo tre cifre.

> **Note**
>
> La notazione simbolica, ad esempio, non **u+x** è supportata per l'impostazione. `mode`

Esempi:

- `mode: "0644"`fornisce autorizzazioni di lettura e scrittura al proprietario dell'oggetto (6), autorizzazioni di sola lettura al gruppo (4) e autorizzazioni di sola lettura a tutti gli altri utenti (4).
- `mode: 644`concede le stesse autorizzazioni di. `mode: "0644"`
- `mode: 4755`imposta l'attributo setuid (4), fornisce i permessi di controllo completo al proprietario (7), fornisce i permessi di lettura ed esecuzione al gruppo (5) e fornisce i permessi di lettura ed esecuzione a tutti gli altri utenti (5).

  Per altri esempi, consultate la documentazione del comando Linux chmod.

  Se la modalità non è specificata, tutte le modalità esistenti applicate alla struttura originale del file o della cartella rimangono invariate dopo l'operazione di copia.

- `acls`: opzionale. Elenco di stringhe di caratteri che rappresenta una o più voci della lista di controllo accessi (ACL) applicate a `object`. Ad esempio, **`u:bob:rw`** rappresenta le autorizzazioni di lettura e scrittura per l'utente **bob**. Per ulteriori informazioni, consulta gli esempi del formato per le voci ACL nella documentazione del comando `setfacl` di Linux. Puoi specificare più voci ACL. Se non `acls` è specificata, gli eventuali ACL esistenti applicati al file o alla directory/folder struttura originale rimangono invariati dopo l'operazione di copia. Eventuali liste di controllo degli accessi esistenti vengono sostituite.

  Specifica `acls` con un trattino (-) seguito da uno spazio e quindi da una stringa, ad esempio, `- u:jane:rw`. Se hai più liste ACL, ogni lista viene specificata su una riga separata.

> **Note**
>
> L'impostazione di utenti senza nome, gruppi senza nome o altre voci ACL simili causa l'errore del file. AppSpec Utilizza `mode` per specificare i tipi di autorizzazione.

- `context`: opzionale. Per le istanze abilitate per Security-Enhanced Linux (SELinux), un elenco di etichette di contesto rilevanti per la sicurezza da applicare agli oggetti copiati. Le etichette vengono specificate come chiavi contenenti `user`, `type` e `range`. Per ulteriori informazioni, consulta la documentazione di SELinux. Ogni chiave viene specificata con una stringa. Se non specificato, le etichette esistenti applicate al file o alla directory/folder struttura originale rimangono invariate dopo l'operazione di copia.
  - `user`: opzionale. L'utente SELinux.
  - `type`: opzionale. Il nome del tipo SELinux.
  - `range`: opzionale. Lo specificatore dell'intervallo di SELinux. Ciò non ha effetto a meno che Multi-Level Security (MLS) e Multi-Category Security (MCS) non siano abilitate sulla macchina. Se non è abilitata, l'impostazione di `range` è predefinita su **`s0`**.

  Specificare `context` con una stringa (ad esempio, `user: unconfined_u`). Ogni `context` viene specificato su una riga separata.
- `type`: opzionale. I tipi di oggetti a cui si applicano le autorizzazioni specificate. `type` è una stringa che può essere impostata su **`file`** o **`directory`**. Se **`file`** è specificato, le autorizzazioni vengono applicate solo ai file contenuti direttamente in `object` dopo l'operazione di copia (e non a `object` stesso). Se **`directory`** specificato, le autorizzazioni vengono applicate in modo ricorsivo a tutto ciò directories/folders che si trova in qualsiasi punto `object` dopo l'operazione di copia (ma non a se stesse). `object`

  Specifica `type` con un trattino (-) seguito da uno spazio e quindi da una stringa, ad esempio, `- file`.

Esempio di sezione 'Autorizzazioni'

L'esempio seguente mostra come specificare la sezione `'permissions'` con le istruzioni `object`, `pattern`, `except`, `owner`, `mode` e `type`. Questo esempio si applica solo alle istanze Amazon Linux, Ubuntu Server e RHEL. In questo esempio presupponiamo che i file e le cartelle seguenti siano copiati nell'istanza in questa gerarchia:

```
/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
```

Il AppSpec file seguente mostra come impostare le autorizzazioni per questi file e cartelle dopo che sono stati copiati:

```
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
```

Le autorizzazioni risultanti sono le seguenti:

```
-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
```

L'esempio seguente mostra come specificare la sezione `'permissions'` con l'aggiunta delle istruzioni `context` e `acls`. Questo esempio si applica solo alle istanze Amazon Linux, Ubuntu Server e RHEL.

```
permissions:
  - object: /var/www/html/WordPress
    pattern: "**"
    except: [/var/www/html/WordPress/ReadMe.txt]
    owner: bob
    group: writers
    mode: 644
    acls:
      - u:mary:rw
      - u:sam:rw
      - m::rw
```

```
    context:
      user: unconfined_u
      type: httpd_sys_content_t
      range: s0
    type:
      - file
```

AppSpec sezione 'hooks'

Il contenuto nella `'hooks'` sezione del AppSpec file varia a seconda della piattaforma di elaborazione utilizzata per la distribuzione. La sezione `'hooks'` per una distribuzione EC2/On-Premises contiene le mappature che collegano gli hook degli eventi del ciclo di vita della distribuzione a uno o più script. La `'hooks'` sezione relativa a una distribuzione Lambda o Amazon ECS specifica le funzioni di convalida Lambda da eseguire durante un evento del ciclo di vita della distribuzione. Se l'hook di un evento non è presente, non viene eseguita alcuna operazione per quell'evento. Questa sezione è obbligatoria solo se esegui script o funzioni di convalida Lambda come parte della distribuzione.

Argomenti

- AppSpec sezione 'hook' per una distribuzione Amazon ECS
- AppSpec sezione 'hooks' per un AWS Implementazione Lambda
- AppSpec sezione 'hooks' per una distribuzione EC2/On-Premises

AppSpec sezione 'hook' per una distribuzione Amazon ECS

Argomenti

- Elenco degli hook relativi agli eventi del ciclo di vita per una distribuzione Amazon ECS
- Esegui l'ordine degli hook in una distribuzione Amazon ECS.
- Struttura della sezione «ganci»
- Esempio di funzione «ganci» Lambda

Elenco degli hook relativi agli eventi del ciclo di vita per una distribuzione Amazon ECS

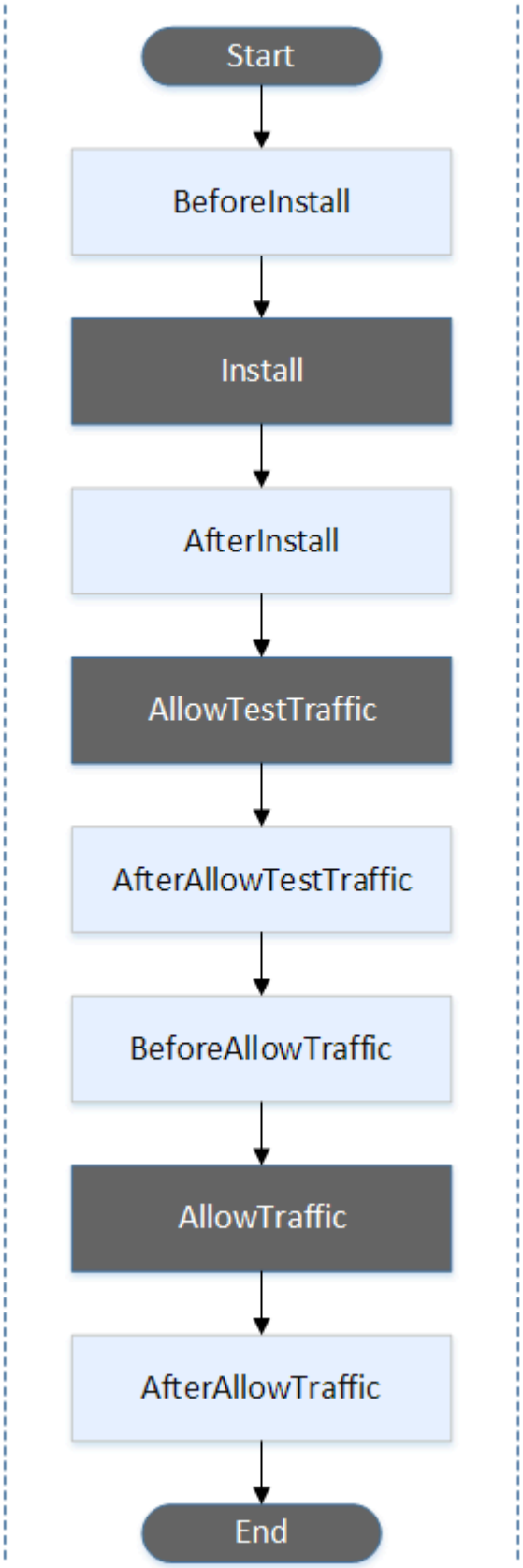
Un hook AWS Lambda è una funzione Lambda specificata con una stringa su una nuova riga dopo il nome dell'evento del ciclo di vita. Ogni hook viene eseguito una volta per distribuzione. Di seguito sono riportate le descrizioni degli eventi del ciclo di vita in cui è possibile eseguire un hook durante una distribuzione Amazon ECS.

- `BeforeInstall`— Da utilizzare per eseguire attività prima della creazione del set di attività sostitutivo. Un gruppo target è associato al set di attività originale. Se un listener di test facoltativo è specificato, esso viene associato al set di attività originale. Un rollback non è possibile in questa fase.
- `AfterInstall`— Da utilizzare per eseguire attività dopo la creazione del set di attività sostitutivo e l'associazione di uno dei gruppi target. Se un listener di test facoltativo è specificato, esso viene associato al set di attività originale. I risultati di una funzione di hook per questo evento del ciclo di vita possono attivare un rollback.
- `AfterAllowTestTraffic`— Da utilizzare per eseguire attività dopo che il listener di test ha inviato il traffico al set di attività sostitutivo. I risultati di una funzione di hook a questo punto possono attivare un rollback.
- `BeforeAllowTraffic`— Da utilizzare per eseguire attività dopo che il secondo gruppo target è stato associato al set di attività sostitutivo, ma prima che il traffico venga spostato verso il set di attività sostitutivo. I risultati di una funzione di hook per questo evento del ciclo di vita possono attivare un rollback.
- `AfterAllowTraffic`— Da utilizzare per eseguire attività dopo che il secondo gruppo target ha inviato traffico al set di attività sostitutivo. I risultati di una funzione di hook per questo evento del ciclo di vita possono attivare un rollback.

Per ulteriori informazioni, consulta Cosa succede durante una distribuzione di Amazon ECS e Tutorial: distribuisci un servizio Amazon ECS con un test di convalida.

Esegui l'ordine degli hook in una distribuzione Amazon ECS.

In una distribuzione Amazon ECS, gli event hook vengono eseguiti nel seguente ordine:


Start
BeforeInstall
Install
AfterInstall
AllowTestTraffic
AfterAllowTestTraffic
BeforeAllowTraffic
AllowTraffic
AfterAllowTraffic
End

> **Note**
>
> Gli eventi di avvio TestTrafficAllowTraffic, installazione e fine della distribuzione non possono essere gestiti tramite script, motivo per cui appaiono in grigio in questo diagramma.

Struttura della sezione «ganci»

Di seguito sono riportati alcuni esempi della struttura della sezione `'hooks'`.

Utilizzando YAML:

```
Hooks:
  - BeforeInstall: "BeforeInstallHookFunctionName"
  - AfterInstall: "AfterInstallHookFunctionName"
  - AfterAllowTestTraffic: "AfterAllowTestTrafficHookFunctionName"
  - BeforeAllowTraffic: "BeforeAllowTrafficHookFunctionName"
  - AfterAllowTraffic: "AfterAllowTrafficHookFunctionName"
```

Utilizzando JSON:

```
"Hooks": [
  {
   "BeforeInstall": "BeforeInstallHookFunctionName"
  },
  {
   "AfterInstall": "AfterInstallHookFunctionName"
  },
  {
   "AfterAllowTestTraffic": "AfterAllowTestTrafficHookFunctionName"
  },
  {
   "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
  },
  {
   "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
  }
 ]
}
```

Esempio di funzione «ganci» Lambda

Utilizza la `'hooks'` sezione per specificare una funzione Lambda che CodeDeploy può chiamare per convalidare una distribuzione Amazon ECS. Puoi utilizzare la stessa funzione o una diversa per gli eventi del ciclo di vita della `AfterAllowTraffic` distribuzione `BeforeInstall` `AfterInstallAfterAllowTestTraffic`,`BeforeAllowTraffic`, e. Dopo il completamento dei test di convalida, la funzione `AfterAllowTraffic` Lambda CodeDeploy richiama e fornisce il risultato `Succeeded` di o. `Failed`

> **Important**
>
> L'implementazione è considerata fallita se non CodeDeploy viene notificata dalla funzione di convalida Lambda entro un'ora.

Prima di richiamare una funzione Lambda hook, il server deve ricevere una notifica dell'ID di distribuzione e dell'ID di esecuzione dell'hook dell'evento lifecycle utilizzando il comando. `putLifecycleEventHookExecutionStatus`

Di seguito è riportato un esempio di funzione Lambda hook scritta in. Node.js

```
'use strict';

const aws = require('aws-sdk');
const codedeploy = new aws.CodeDeploy({apiVersion: '2014-10-06'});

exports.handler = (event, context, callback) => {
    //Read the DeploymentId from the event payload.
    var deploymentId = event.DeploymentId;

    //Read the LifecycleEventHookExecutionId from the event payload
    var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

    /*
     Enter validation tests here.
    */

    // Prepare the validation test results with the deploymentId and
    // the lifecycleEventHookExecutionId for CodeDeploy.
    var params = {
        deploymentId: deploymentId,
```

```
        lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
        status: 'Succeeded' // status can be 'Succeeded' or 'Failed'
    };

    // Pass CodeDeploy the prepared validation test results.
    codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
        if (err) {
            // Validation failed.
            callback('Validation test failed');
        } else {
            // Validation succeeded.
            callback(null, 'Validation test succeeded');
        }
    });
};
```

AppSpec sezione 'hooks' per un AWS Implementazione Lambda

Argomenti

- Elenco degli hook relativi agli eventi del ciclo di vita per un AWS Implementazione Lambda
- Esegui l'ordine degli hook in una distribuzione della versione della funzione Lambda
- Struttura della sezione «ganci»
- Esempio di funzione «ganci» Lambda

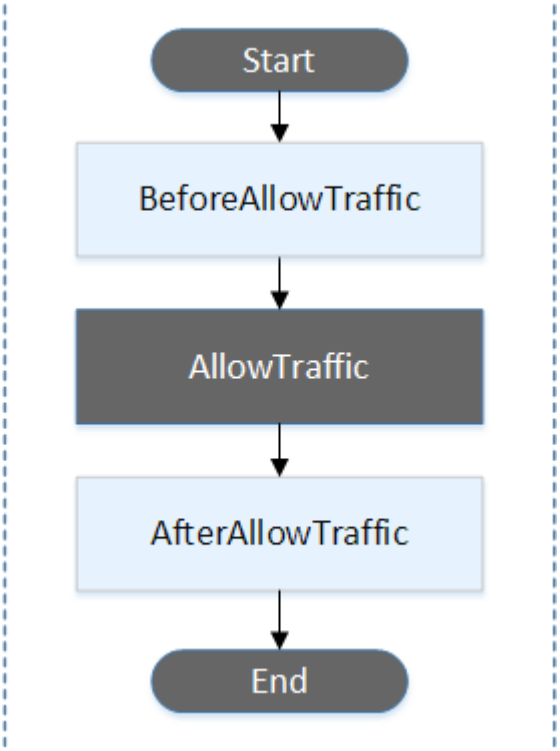
Elenco degli hook relativi agli eventi del ciclo di vita per un AWS Implementazione Lambda

Un hook AWS Lambda è una funzione Lambda specificata con una stringa su una nuova riga dopo il nome dell'evento del ciclo di vita. Ogni hook viene eseguito una volta per distribuzione. Di seguito sono riportate le descrizioni degli hook disponibili per l'uso nel file. AppSpec

- BeforeAllowTraffic— Da utilizzare per eseguire attività prima che il traffico venga spostato alla versione della funzione Lambda implementata.
- AfterAllowTraffic— Da utilizzare per eseguire attività dopo che tutto il traffico è stato spostato sulla versione della funzione Lambda implementata.

Esegui l'ordine degli hook in una distribuzione della versione della funzione Lambda

In una versione di una funzione Lambda senza server, gli event hook vengono eseguiti nel seguente ordine:



> **Note**
>
> Gli eventi di inizio e fine della distribuzione non possono essere gestiti tramite script, motivo per cui appaiono in grigio in questo diagramma. AllowTraffic

Struttura della sezione «ganci»

Di seguito sono riportati alcuni esempi della struttura della sezione "hooks".

Utilizzando YAML:

```
hooks:
   - BeforeAllowTraffic: BeforeAllowTrafficHookFunctionName
   - AfterAllowTraffic: AfterAllowTrafficHookFunctionName
```

Utilizzando JSON:

```
"hooks": [{
    "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
    },
    {
```

```
    "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
}]
```

Esempio di funzione «ganci» Lambda

Usa la sezione 'hooks' per specificare una funzione Lambda CodeDeploy che può chiamare per convalidare una distribuzione Lambda. È possibile utilizzare la stessa funzione o una diversa per gli eventi del ciclo di vita della distribuzione`BeforeAllowTraffic`. `AfterAllowTraffic` Dopo il completamento dei test di convalida, la funzione di convalida Lambda CodeDeploy richiama e fornisce il risultato di o. `Succeeded` `Failed`

> **Important**
>
> L'implementazione è considerata fallita se non CodeDeploy viene notificata dalla funzione di convalida Lambda entro un'ora.

Prima di richiamare una funzione Lambda hook, il server deve ricevere una notifica dell'ID di distribuzione e dell'ID di esecuzione dell'hook dell'evento lifecycle utilizzando il comando. `putLifecycleEventHookExecutionStatus`

Di seguito è riportato un esempio di funzione Lambda hook scritta in. Node.js

```
'use strict';

const aws = require('aws-sdk');
const codedeploy = new aws.CodeDeploy({apiVersion: '2014-10-06'});

exports.handler = (event, context, callback) => {
    //Read the DeploymentId from the event payload.
    var deploymentId = event.DeploymentId;

    //Read the LifecycleEventHookExecutionId from the event payload
    var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

    /*
     Enter validation tests here.
    */

    // Prepare the validation test results with the deploymentId and
    // the lifecycleEventHookExecutionId for CodeDeploy.
```

```
    var params = {
        deploymentId: deploymentId,
        lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
        status: 'Succeeded' // status can be 'Succeeded' or 'Failed'
    };

    // Pass CodeDeploy the prepared validation test results.
    codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
        if (err) {
            // Validation failed.
            callback('Validation test failed');
        } else {
            // Validation succeeded.
            callback(null, 'Validation test succeeded');
        }
    });
};
```

AppSpec sezione 'hooks' per una distribuzione EC2/On-Premises

Argomenti

- Elenco degli hook relativi agli eventi del ciclo di vita
- Disponibilità di Lifecycle Event Hook
- Esegui l'ordine degli hook in una distribuzione
- Struttura della sezione «ganci»
- Riferimento ai file negli script hook
- Disponibilità delle variabili di ambiente per gli hook
- Esempio di hook

Elenco degli hook relativi agli eventi del ciclo di vita

Un hook EC2/On-Premises di distribuzione viene eseguito una volta per distribuzione su un'istanza. Puoi specificare uno o più script da eseguire in un hook. Ogni hook per un evento del ciclo di vita viene specificato con una stringa su una riga separata. Di seguito sono riportate le descrizioni degli hook disponibili per l'uso nel AppSpec file.

Per informazioni sugli hook degli eventi del ciclo di vita validi per i vari tipi di rollback e di distribuzione, consulta Disponibilità di Lifecycle Event Hook.

- `ApplicationStop`— Questo evento del ciclo di vita della distribuzione si verifica anche prima del download della revisione dell'applicazione. Puoi specificare gli script per questo evento per arrestare correttamente l'applicazione o rimuovere i pacchetti attualmente installati in preparazione di una distribuzione. Il AppSpec file e gli script utilizzati per questo evento del ciclo di vita della distribuzione provengono dalla precedente revisione dell'applicazione distribuita correttamente.

  > **Note**
  >
  > Un AppSpec file non esiste su un'istanza prima della distribuzione su di essa. Per questo motivo, l'hook `ApplicationStop` non viene eseguito la prima volta che lo distribuisci all'istanza. Puoi utilizzare l'hook `ApplicationStop` la seconda volta che distribuisci a un'istanza.

  Per determinare la posizione dell'ultima revisione dell'applicazione distribuita correttamente, l' CodeDeploy agente cerca la posizione elencata nel file. *`deployment-group-id`*`_last_successful_install` Questo file si trova nella:

  `/opt/codedeploy-agent/deployment-root/deployment-instructions`cartella su istanze Amazon Linux, Ubuntu Server e RHEL Amazon EC2.

  `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions`cartella su istanze Amazon EC2 di Windows Server.

  Per risolvere i problemi di una distribuzione che non riesce durante l'evento del ciclo di vita della distribuzione `ApplicationStop`, consulta Risoluzione dei problemi relativi a un evento non riuscito o ApplicationStop AfterBlockTraffic relativo al BeforeBlockTraffic ciclo di vita dell'implementazione.

- `DownloadBundle`— Durante questo evento del ciclo di vita della distribuzione, l' CodeDeploy agente copia i file di revisione dell'applicazione in una posizione temporanea:

  `/opt/codedeploy-agent/deployment-root/`*`deployment-group-id`*`/`*`deployment-id`*`/deployment-archive`cartella su istanze Amazon Linux, Ubuntu Server e RHEL Amazon EC2.

  `C:\ProgramData\Amazon\CodeDeploy\`*`deployment-group-id`*`\`*`deployment-id`*`\deployment-archive`cartella su istanze Amazon EC2 di Windows Server.

  Questo evento è riservato all' CodeDeploy agente e non può essere utilizzato per eseguire script.

Per risolvere i problemi di una distribuzione che non riesce durante l'evento del ciclo di vita della distribuzione `DownloadBundle`, consulta Risoluzione dei problemi relativi a un evento del ciclo di vita di DownloadBundle distribuzione non riuscito con UnknownError: not open-for reading.

- `BeforeInstall`— È possibile utilizzare questo evento del ciclo di vita della distribuzione per attività di preinstallazione, come la decrittografia dei file e la creazione di un backup della versione corrente.
- `Install`— Durante questo evento del ciclo di vita della distribuzione, l' CodeDeployagente copia i file di revisione dalla posizione temporanea alla cartella di destinazione finale. Questo evento è riservato all' CodeDeploy agente e non può essere utilizzato per eseguire script.
- `AfterInstall`— È possibile utilizzare questo evento del ciclo di vita della distribuzione per attività come la configurazione dell'applicazione o la modifica delle autorizzazioni dei file.
- `ApplicationStart`— In genere si utilizza questo evento del ciclo di vita della distribuzione per riavviare i servizi che sono stati interrotti durante l'operazione. `ApplicationStop`
- `ValidateService`— Questo è l'ultimo evento del ciclo di vita della distribuzione. È utilizzato per verificare se la distribuzione è stata completata.
- `BeforeBlockTraffic`— È possibile utilizzare questo evento del ciclo di vita della distribuzione per eseguire attività sulle istanze prima che vengano cancellate da un sistema di bilanciamento del carico.

  Per risolvere i problemi di una distribuzione che non riesce durante l'evento del ciclo di vita della distribuzione `BeforeBlockTraffic`, consulta Risoluzione dei problemi relativi a un evento non riuscito o ApplicationStop AfterBlockTraffic relativo al BeforeBlockTraffic ciclo di vita dell'implementazione.
- `BlockTraffic`— Durante questo evento del ciclo di vita della distribuzione, al traffico Internet viene impedito l'accesso alle istanze che attualmente servono traffico. Questo evento è riservato all' CodeDeploy agente e non può essere utilizzato per eseguire script.
- `AfterBlockTraffic`— È possibile utilizzare questo evento del ciclo di vita della distribuzione per eseguire attività sulle istanze dopo che queste sono state cancellate dal rispettivo sistema di bilanciamento del carico.

  Per risolvere i problemi di una distribuzione che non riesce durante l'evento del ciclo di vita della distribuzione `AfterBlockTraffic`, consulta Risoluzione dei problemi relativi a un evento non riuscito o ApplicationStop AfterBlockTraffic relativo al BeforeBlockTraffic ciclo di vita dell'implementazione.

- `BeforeAllowTraffic`— È possibile utilizzare questo evento del ciclo di vita della distribuzione per eseguire attività sulle istanze prima che vengano registrate con un sistema di bilanciamento del carico.
- `AllowTraffic`— Durante questo evento del ciclo di vita della distribuzione, il traffico Internet può accedere alle istanze dopo una distribuzione. Questo evento è riservato all' CodeDeploy agente e non può essere utilizzato per eseguire script.
- `AfterAllowTraffic`— È possibile utilizzare questo evento del ciclo di vita della distribuzione per eseguire attività sulle istanze dopo che sono state registrate con un sistema di bilanciamento del carico.

Disponibilità di Lifecycle Event Hook

La tabella seguente elenca gli hook degli eventi del ciclo di vita disponibili per ogni scenario di distribuzione e rollback.

Nome dell'evento del ciclo di vita	Implementazione del lancio di Auto Scaling[1]	Implementazione della terminazione con Auto Scaling[1]	In-place distribuzione[2]	Blue/green distribuzione: istanze originali	Blue/green distribuzione: istanze sostitutive	Blue/green rollback di implementazione: istanze originali	Blue/green rollback di implementazione: istanze sostitutive
ApplicationStop	✓	✓	✓		✓		
DownloadBundle[3]	✓		✓		✓		
BeforeInstall	✓		✓		✓		
Installa [3]	✓		✓		✓		
AfterInstall	✓		✓		✓		

Nome dell'evento del ciclo di vita	Implementazione del lancio di Auto Scaling[1]	Implementazione della terminazione con Auto Scaling[1]	In-place distribuzione[2]	Blue/green distribuzione: istanze originali	Blue/green distribuzione: istanze sostitutive	Blue/green rollback di implementazione: istanze originali	Blue/green rollback di implementazione: istanze sostitutive
ApplicationStart	✓		✓		✓		
ValidateService	✓		✓		✓		
BeforeBlockTraffic		✓	✓	✓			✓
BlockTraffic[3]		✓	✓	✓			✓
AfterBlockTraffic		✓	✓	✓			✓
BeforeAllowTraffic	✓		✓		✓	✓	
AllowTraffic[3]	✓		✓		✓	✓	
AfterAllowTraffic	✓		✓		✓	✓	

[1] Per informazioni sulle implementazioni di Amazon EC2 Auto Scaling, consulta. Come funziona Amazon EC2 Auto Scaling con CodeDeploy

[2] Si applica anche al rollback di una distribuzione sul posto.

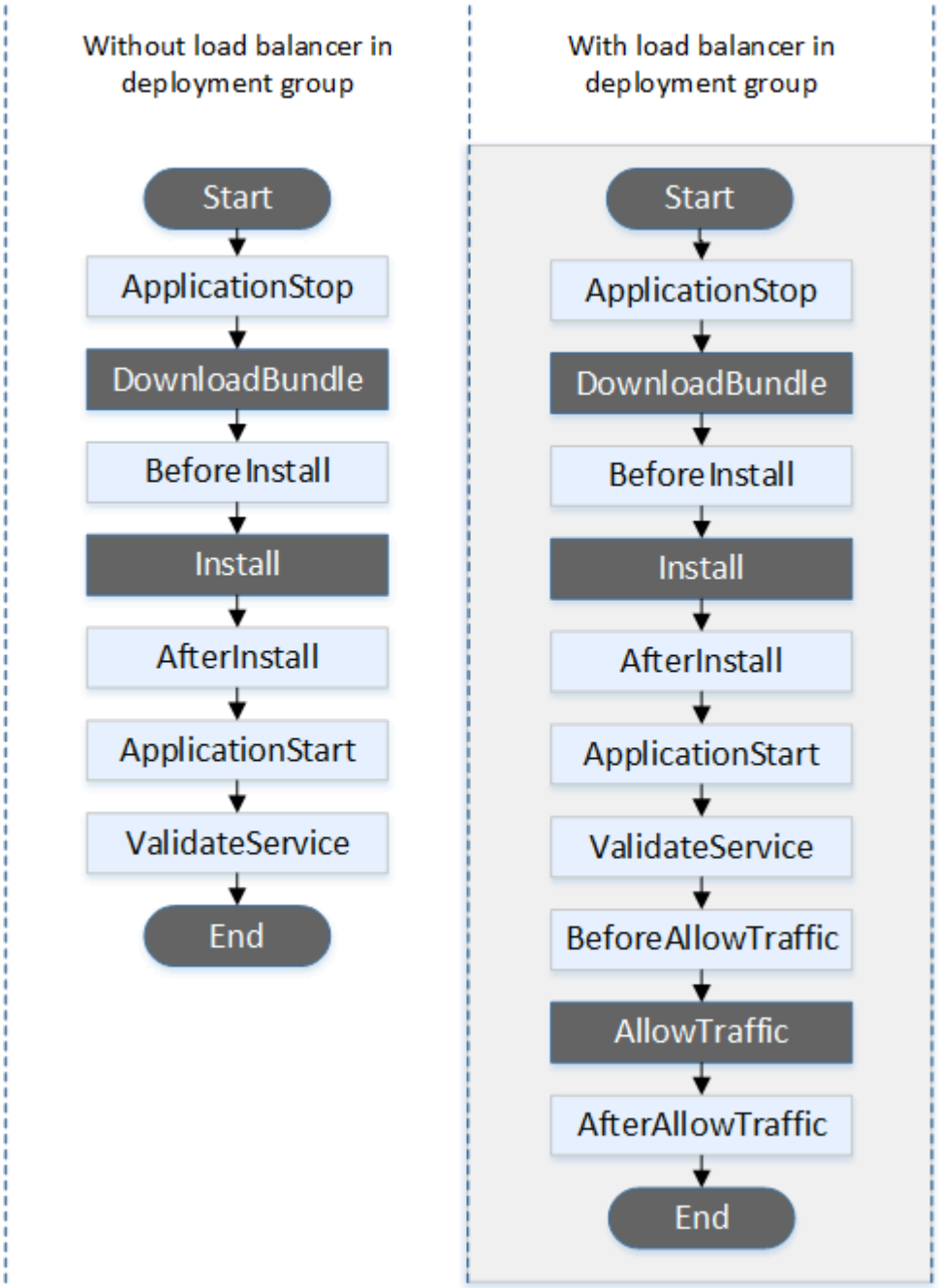
[3] Riservato alle operazioni CodeDeploy . Non può essere utilizzato per l'esecuzione di script.

Esegui l'ordine degli hook in una distribuzione

Implementazioni di lancio di Auto Scaling

Durante una distribuzione con avvio di Auto Scaling, CodeDeploy esegue gli event hook nell'ordine seguente.

Per ulteriori informazioni sulle implementazioni di avvio di Auto Scaling, vedere. Come funziona Amazon EC2 Auto Scaling con CodeDeploy

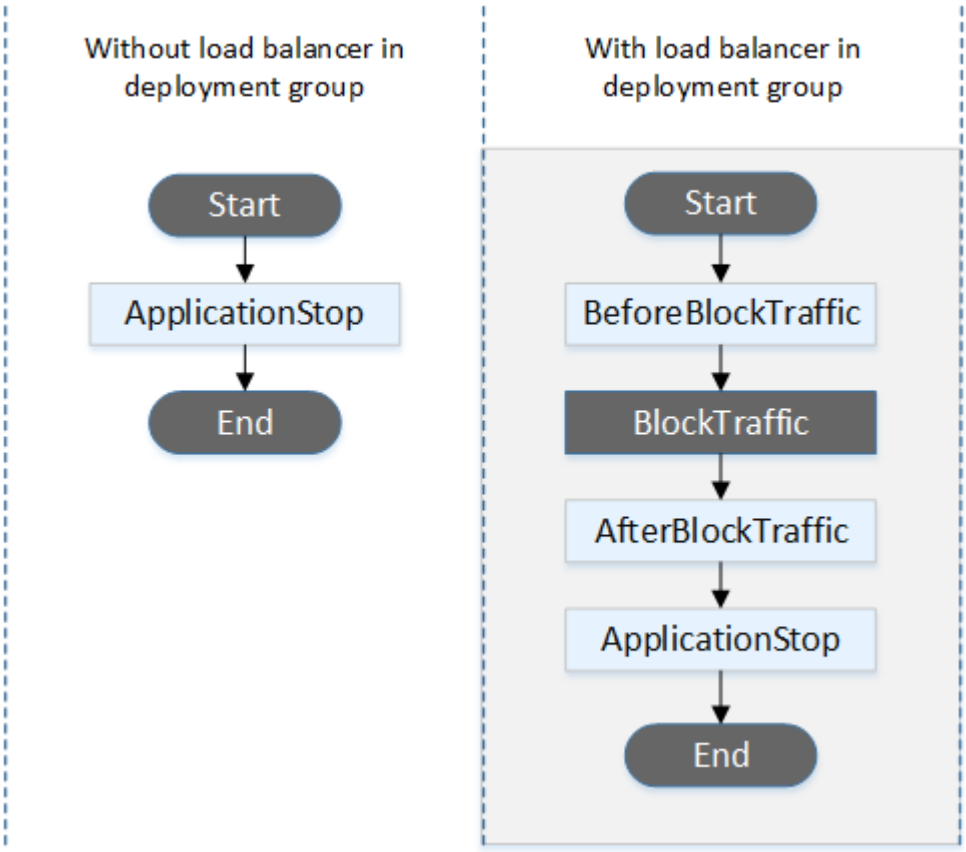


> **Note**
>
> Gli eventi di avvio DownloadBundleAllowTraffic, installazione e fine della distribuzione non possono essere gestiti tramite script, motivo per cui appaiono in grigio in questo diagramma. Tuttavia, è possibile modificare la `'files'` sezione del AppSpec file per specificare cosa viene installato durante l'evento di installazione.

Implementazioni di terminazione con Auto Scaling

Durante una distribuzione con terminazione di Auto Scaling, CodeDeploy esegue gli event hook nell'ordine seguente.

Per ulteriori informazioni sulle implementazioni di terminazione con Auto Scaling, vedere. Abilitazione delle implementazioni di terminazione durante gli eventi scale-in di Auto Scaling

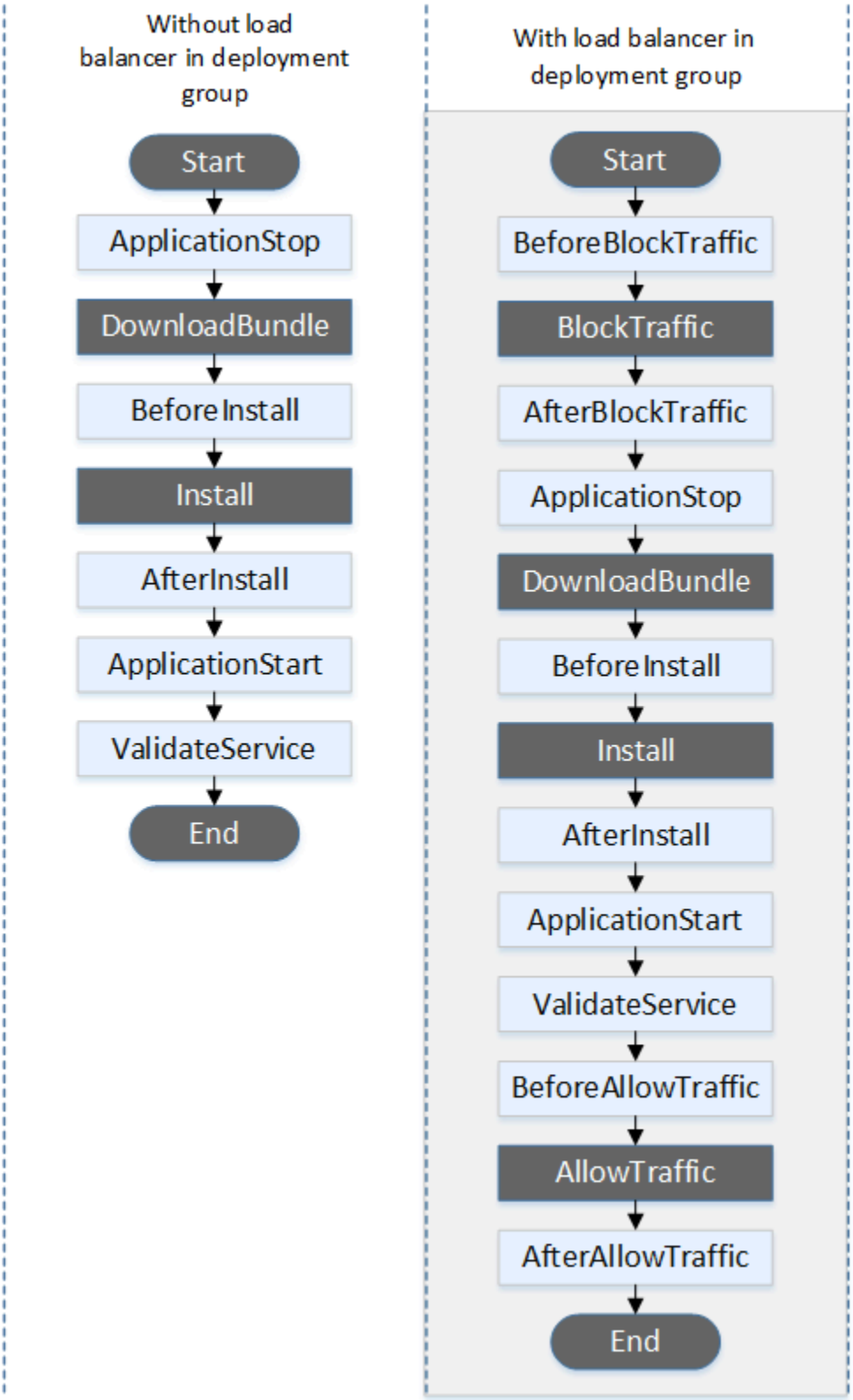


> **Note**
>
> Gli eventi di inizio e fine della distribuzione non possono essere gestiti tramite script, motivo per cui appaiono in grigio in questo diagramma. BlockTraffic

In-place distribuzioni

In una distribuzione in loco, incluso il rollback di una distribuzione in loco, gli hook degli eventi vengono eseguiti nel seguente ordine:

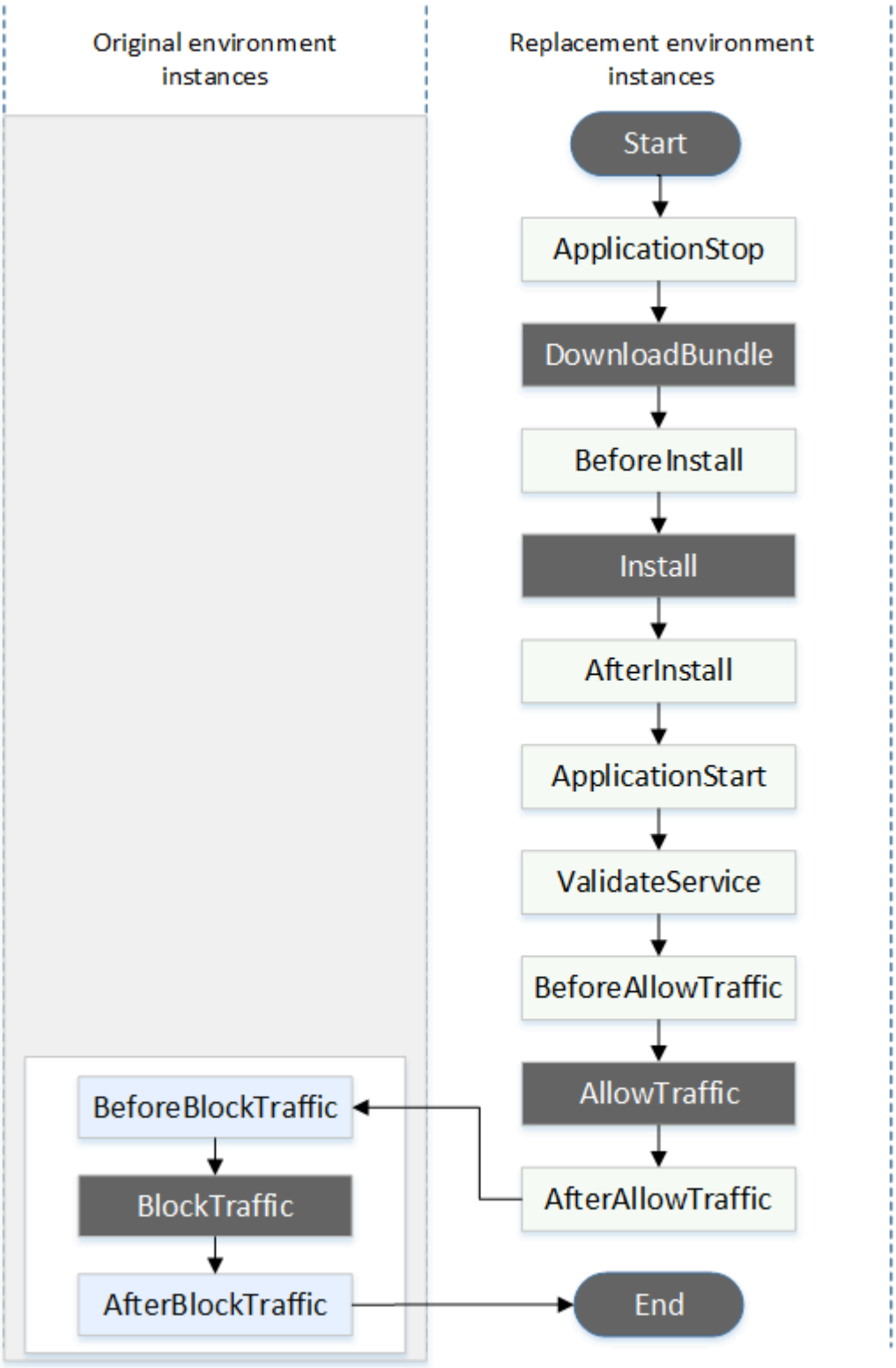
> **Note**
>
> Per le distribuzioni sul posto, i sei hook relativi al blocco e all'autorizzazione del traffico si applicano solo se si specifica un Classic Load Balancer, Application Load Balancer o Network Load Balancer di Elastic Load Balancing nel gruppo di distribuzione.


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**
>
> Gli eventi di avvio DownloadBundle, installazione e fine della distribuzione non possono essere gestiti tramite script, motivo per cui appaiono in grigio in questo diagramma. Tuttavia, è possibile modificare la `'files'` sezione del AppSpec file per specificare cosa viene installato durante l'evento di installazione.

Blue/green distribuzioni

In una blue/green distribuzione, gli event hook vengono eseguiti nel seguente ordine:


Original environment instances
Replacement environment instances
Start
ApplicationStop
DownloadBundle
BeforeInstall
Install
AfterInstall
ApplicationStart
ValidateService
BeforeAllowTraffic
AllowTraffic
AfterAllowTraffic
BeforeBlockTraffic
BlockTraffic
AfterBlockTraffic
End

> **Note**
>
> Gli eventi di avvio DownloadBundleBlockTrafficAllowTraffic, installazione e fine della distribuzione non possono essere gestiti tramite script, motivo per cui vengono visualizzati in grigio in questo diagramma. Tuttavia, è possibile modificare la sezione 'files' del AppSpec file per specificare cosa viene installato durante l'evento di installazione.

Struttura della sezione «ganci»

La sezione `'hooks'` ha la seguente struttura:

```
hooks:
   deployment-lifecycle-event-name:
     - location: script-location
       timeout: timeout-in-seconds
       runas: user-name
```

Puoi includere i seguenti elementi in una voce hook dopo il nome dell'evento del ciclo di vita della distribuzione:

location

Obbligatorio. La posizione nel bundle del file di script per la revisione. La posizione degli script specificati nella `hooks` sezione è relativa alla radice del pacchetto di revisione dell'applicazione. Per ulteriori informazioni, consulta Pianifica una revisione per CodeDeploy.

timeout

Opzionale. Il numero di secondi dedicati all'esecuzione dello script prima che venga considerato non riuscito. Il valore predefinito è 3600 secondi (1 ora).

> **Note**
>
> 3600 secondi (1 ora) è il periodo di tempo massimo consentito per l'esecuzione di uno script per ogni evento del ciclo di vita della distribuzione. Se gli script superano questo limite, la distribuzione viene interrotta e la distribuzione nell'istanza non riesce. Assicurati che il numero totale di secondi specificato per timeout per tutti gli script in ogni evento del ciclo di vita della distribuzione non superi questo limite.

runas

Opzionale. L'utente da rappresentare quando viene eseguito lo script. Per impostazione predefinita, questo è l' CodeDeploy agente in esecuzione sull'istanza. CodeDeploy non memorizza le password, quindi l'utente non può essere impersonato se l'utente runas necessita di una password. Questo elemento si applica solo alle istanze di Amazon Linux e Ubuntu Server.

Riferimento ai file negli script hook

Se stai collegando uno script a un evento CodeDeploy del ciclo di vita come descritto in AppSpec sezione 'hooks' e desideri fare riferimento a un file (ad esempio,`helper.sh`) nello script, dovrai specificare utilizzando: `helper.sh`

- (Consigliato) Un percorso assoluto. Consulta Utilizzo di percorsi assoluti.
- Un percorso relativo. Consulta Utilizzo di percorsi relativi.

Utilizzo di percorsi assoluti

Per fare riferimento a un file usando il suo percorso assoluto, puoi:

- Specificate il percorso assoluto `files` nella sezione del AppSpec file, nella `destination` proprietà. Quindi, specifica lo stesso percorso assoluto nel tuo script hook. Per ulteriori informazioni, consulta AppSpec sezione 'file' (solo distribuzioni) EC2/On-Premises.
- Specificate un percorso assoluto dinamico nello script hook. Per ulteriori informazioni, consulta Deployment archive location.

Posizione dell'archivio di distribuzione

Durante l'evento del DownloadBundleciclo di vita, l' CodeDeploy agente estrae la revisione per la distribuzione in una directory con il seguente formato:

`root-directory``/`*`deployment-group-id`*`/`*`deployment-id`*`/deployment-archive`

La *`root-directory`* parte del percorso è sempre impostata sul valore predefinito mostrato nella tabella seguente o è controllata dall'impostazione di configurazione. `:root_dir` Per ulteriori informazioni sulle impostazioni di configurazione, vedereCodeDeploy riferimento alla configurazione dell'agente.

Piattaforma Agent	Directory principale predefinita
Linux: tutte le distribuzioni rpm	`/opt/codedeploy-agent/deployment-root`
Ubuntu Server: tutte le distribuzioni deb	`/opt/codedeploy-agent/deployment-root`
Windows Server	`%ProgramData%\Amazon\CodeDeploy`

Dai tuoi script hook, puoi accedere all'archivio di distribuzione corrente utilizzando il percorso della directory principale e le variabili di ambiente `DEPLOYMENT_ID` and`DEPLOYMENT_GROUP_ID`. Per ulteriori informazioni sulle variabili che è possibile utilizzare, vedereDisponibilità delle variabili di ambiente per gli hook.

Ad esempio, ecco come è possibile accedere a un `data.json` file che si trova nella radice della revisione su Linux:

```
#!/bin/bash

rootDirectory="/opt/codedeploy-agent/deployment-root" # note: this will be different if
 you
                                                      # customize the :root_dir
 configuration
dataFile="$rootDirectory/$DEPLOYMENT_GROUP_ID/$DEPLOYMENT_ID/deployment-archive/
data.json"
data=$(cat dataFile)
```

Come altro esempio, ecco come è possibile accedere a un `data.json` file che si trova alla radice della revisione utilizzando Powershell su Windows:

```
$rootDirectory="$env:ProgramData\Amazon\CodeDeploy" # note: this will be different if
 you
                                                    # customize the :root_dir
 configuration
$dataFile="$rootDirectory\$env:DEPLOYMENT_GROUP_ID\$env:DEPLOYMENT_ID\deployment-
archive\data.json"
$data=(Get-Content $dataFile)
```

Utilizzo di percorsi relativi

Per fare riferimento a un file utilizzando il relativo percorso, è necessario conoscere la directory di lavoro dell' CodeDeploy agente. I percorsi dei file sono relativi a questa directory.

La tabella seguente mostra la directory di lavoro per ogni piattaforma supportata dall' CodeDeploy agente.

Piattaforma dell'agente	Metodo di gestione dei processi	Directory di lavoro per gli script degli eventi del ciclo di vita
Linux: tutte le distribuzioni rpm	systemd (impostazione predefinita)	`/`
	init.d — Scopri di più	`/opt/codedeploy-agent`
Ubuntu Server: tutte le distribuzioni debian	tutto	`/opt/codedeploy-agent`
Windows Server	non applicabile	`C:\Windows\System32`

Disponibilità delle variabili di ambiente per gli hook

Durante ogni evento del ciclo di vita della distribuzione, gli script di hook possono accedere alle seguenti variabili di ambiente:

APPLICATION_NAME

Il nome dell'applicazione CodeDeploy che fa parte della distribuzione corrente (ad esempio,`WordPress_App`).

DEPLOYMENT_ID

L'ID CodeDeploy è stato assegnato alla distribuzione corrente (ad esempio,`d-AB1CDEF23`).

DEPLOYMENT_GROUP_NAME

Il nome del gruppo di distribuzione CodeDeploy che fa parte della distribuzione corrente (ad esempio,`WordPress_DepGroup`).

DEPLOYMENT_GROUP_ID

L'ID del gruppo di distribuzione CodeDeploy che fa parte della distribuzione corrente (ad esempio,`b1a2189b-dd90-4ef5-8f40-4c1c5EXAMPLE`).

LIFECYCLE_EVENT

Il nome dell'evento del ciclo di vita della distribuzione corrente (ad esempio `AfterInstall`).

Queste variabili di ambiente sono locali per ciascun evento del ciclo di vita della distribuzione.

Sono disponibili variabili di ambiente aggiuntive per agganciare gli script a seconda dell'origine del pacchetto di distribuzione:

Pacchetto da Amazon S3

- BUNDLE_BUCKET

  Il nome del bucket Amazon S3 da cui è stato scaricato il pacchetto di distribuzione (ad esempio,). `my-s3-bucket`
- BUNDLE_KEY

  La chiave oggetto per il pacchetto scaricato all'interno del bucket Amazon S3 (ad esempio,). `WordPress_App.zip`
- BUNDLE_VERSION

  La versione dell'oggetto per il pacchetto (ad esempio,). `3sL4kqtJlcpXroDTDmJ +rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo` Questa variabile viene impostata solo se il bucket Amazon S3 ha abilitato il controllo delle versioni degli oggetti.
- BUNDLE_ETAG

  L'oggetto etag per il pacchetto (ad esempio,). `b10a8db164e0754105b7a99be72e3fe5-4`

Bundle da GitHub

- BUNDLE_COMMIT

  L'hash di commit SHA256 del bundle generato da Git (ad esempio,). `d2a84f4b8b650937ec8f73cd8be2c74add5a911ba64df27458ed8229da804a26`

Il seguente script modifica la porta di ascolto di un server HTTP Apache su 9090 anziché 80 se il valore di DEPLOYMENT_GROUP_NAME è uguale a `Staging`. Questo script devono essere richiamato durante l'evento del ciclo di vita della distribuzione `BeforeInstall`:

```
if [ "$DEPLOYMENT_GROUP_NAME" == "Staging" ]
then
    sed -i -e 's/Listen 80/Listen 9090/g' /etc/httpd/conf/httpd.conf
fi
```

Il seguente esempio di script modifica il livello di verbosità dei messaggi registrati nel log degli errori da avviso a debug se il valore della variabile di ambiente DEPLOYMENT_GROUP_NAME è uguale a `Staging`. Questo script devono essere richiamato durante l'evento del ciclo di vita della distribuzione `BeforeInstall`:

```
if [ "$DEPLOYMENT_GROUP_NAME" == "Staging" ]
then
    sed -i -e 's/LogLevel warn/LogLevel debug/g' /etc/httpd/conf/httpd.conf
fi
```

Il seguente esempio di script sostituisce il testo nella pagina Web specificata con il testo che visualizza il valore di queste variabili di ambiente. Questo script devono essere richiamato durante l'evento del ciclo di vita della distribuzione `AfterInstall`:

```
#!/usr/bin/python

import os

strToSearch="<h2>This application was deployed using CodeDeploy.</h2>"
strToReplace="<h2>This page for "+os.environ['APPLICATION_NAME']+"
 application and "+os.environ['DEPLOYMENT_GROUP_NAME']+" deployment group with
 "+os.environ['DEPLOYMENT_GROUP_ID']+" deployment group ID was generated by a
 "+os.environ['LIFECYCLE_EVENT']+" script during "+os.environ['DEPLOYMENT_ID']+"
 deployment.</h2>"

fp=open("/var/www/html/index.html","r")
buffer=fp.read()
fp.close()

fp=open("/var/www/html/index.html","w")
fp.write(buffer.replace(strToSearch,strToReplace))
fp.close()
```

Esempio di hook

Ecco un esempio della voce hooks che specifica due hook per l'evento del ciclo di vita `AfterInstall`:

```
hooks:
   AfterInstall:
     - location: Scripts/RunResourceTests.sh
       timeout: 180
     - location: Scripts/PostDeploy.sh
       timeout: 180
```

Lo script `Scripts/RunResourceTests.sh` viene eseguito durante la fase `AfterInstall` del processo di distribuzione. La distribuzione non ha esito positivo se lo script impiega più di 180 secondi (3 minuti) per l'esecuzione.

La posizione degli script specificati nella sezione "hooks" è relativa alla radice del bundle di revisione dell'applicazione. Nel precedente esempio, un file denominato `RunResourceTests.sh` si trova in una directory denominata `Scripts`. La directory `Scripts` si trova al livello di radice del bundle. Per ulteriori informazioni, consulta Pianifica una revisione per CodeDeploy.

AppSpec Esempio di file

Questo argomento fornisce AppSpec file di esempio per una AWS Lambda e una EC2/On-Premises distribuzione.

Argomenti

- AppSpec Esempio di file per una distribuzione Amazon ECS
- AppSpec Esempio di file per un AWS Implementazione Lambda
- AppSpec Esempio di file per una distribuzione EC2/On-Premises

AppSpec Esempio di file per una distribuzione Amazon ECS

Ecco un esempio di AppSpec file scritto in YAML per la distribuzione di un servizio Amazon ECS.

```
version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
```

```
      Properties:
        TaskDefinition: "arn:aws:ecs:us-east-1:111222333444:task-definition/my-task-
definition-family-name:1"
        LoadBalancerInfo:
          ContainerName: "SampleApplicationName"
          ContainerPort: 80
# Optional properties
        PlatformVersion: "LATEST"
        NetworkConfiguration:
          AwsvpcConfiguration:
            Subnets: ["subnet-1234abcd","subnet-5678abcd"]
            SecurityGroups: ["sg-12345678"]
            AssignPublicIp: "ENABLED"
        CapacityProviderStrategy:
          - Base: 1
            CapacityProvider: "FARGATE_SPOT"
            Weight: 2
          - Base: 0
            CapacityProvider: "FARGATE"
            Weight: 1
Hooks:
  - BeforeInstall: "LambdaFunctionToValidateBeforeInstall"
  - AfterInstall: "LambdaFunctionToValidateAfterInstall"
  - AfterAllowTestTraffic: "LambdaFunctionToValidateAfterTestTrafficStarts"
  - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeAllowingProductionTraffic"
  - AfterAllowTraffic: "LambdaFunctionToValidateAfterAllowingProductionTraffic"
```

Questa è una versione dell'esempio precedente scritto in formato 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"
        }
    ]
}
```

Ecco è la sequenza degli eventi durante la distribuzione:

1. Prima che l'applicazione Amazon ECS aggiornata venga installata nel set di attività sostitutivo, viene eseguita la funzione `LambdaFunctionToValidateBeforeInstall` Lambda chiamata.
2. Dopo aver installato l'applicazione Amazon ECS aggiornata nel set di attività sostitutivo, ma prima che riceva traffico, viene eseguita la funzione `LambdaFunctionToValidateAfterInstall` Lambda chiamata.
3. Dopo che l'applicazione Amazon ECS sul set di attività sostitutivo inizia a ricevere traffico dal listener di test, viene eseguita la funzione Lambda chiamata. `LambdaFunctionToValidateAfterTestTrafficStarts` È probabile che questa funzione esegua test di convalida per determinare se la distribuzione deve continuare. Se non specifichi un listener di test nel tuo gruppo di distribuzione, questo hook verrà ignorato.
4. Una volta completati tutti i test di convalida nell'`AfterAllowTestTraffic`hook e prima che il traffico di produzione venga inviato all'applicazione Amazon ECS aggiornata, viene eseguita la funzione Lambda chiamata. `LambdaFunctionToValidateBeforeAllowingProductionTraffic`
5. Dopo che il traffico di produzione è stato inviato all'applicazione Amazon ECS aggiornata sul set di attività sostitutivo, viene eseguita la funzione `LambdaFunctionToValidateAfterAllowingProductionTraffic` Lambda chiamata.

Le funzioni Lambda eseguite durante qualsiasi hook possono eseguire test di convalida o raccogliere metriche sul traffico.

AppSpec Esempio di file per un AWS Implementazione Lambda

Ecco un esempio di AppSpec file scritto in YAML per la distribuzione di una versione della funzione Lambda.

```
version: 0.0
Resources:
  - myLambdaFunction:
```

```
      Type: AWS::Lambda::Function
      Properties:
        Name: "myLambdaFunction"
        Alias: "myLambdaFunctionAlias"
        CurrentVersion: "1"
        TargetVersion: "2"
Hooks:
  - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeTrafficShift"
  - AfterAllowTraffic: "LambdaFunctionToValidateAfterTrafficShift"
```

Questa è una versione dell'esempio precedente scritto in formato JSON.

```
{
  "version": 0.0,
  "Resources": [{
   "myLambdaFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
     "Name": "myLambdaFunction",
     "Alias": "myLambdaFunctionAlias",
     "CurrentVersion": "1",
     "TargetVersion": "2"
    }
   }
  }],
  "Hooks": [{
    "BeforeAllowTraffic": "LambdaFunctionToValidateBeforeTrafficShift"
      },
      {
    "AfterAllowTraffic": "LambdaFunctionToValidateAfterTrafficShift"
   }
  ]
 }
```

Ecco è la sequenza degli eventi durante la distribuzione:

1. Prima di spostare il traffico dalla versione 1 di una funzione Lambda `myLambdaFunction` chiamata alla versione 2, esegui una funzione Lambda `LambdaFunctionToValidateBeforeTrafficShift` chiamata che convalida che la distribuzione sia pronta per avviare lo spostamento del traffico.
2. Se `LambdaFunctionToValidateBeforeTrafficShift` ha restituito il codice di uscita 0 (operazione completata), inizia il trasferimento del traffico alla versione 2 di `myLambdaFunction`.

La configurazione della distribuzione per questa distribuzione determina la velocità di trasferimento del traffico.

3. Una volta completato lo spostamento del traffico dalla versione 1 di una funzione Lambda `myLambdaFunction` chiamata alla versione 2, esegui una funzione Lambda `LambdaFunctionToValidateAfterTrafficShift` chiamata che convalida il completamento della distribuzione.

AppSpec Esempio di file per una distribuzione EC2/On-Premises

Ecco un esempio di AppSpec file per una distribuzione locale su un'istanza Amazon Linux, Ubuntu Server o RHEL.

> **Note**
>
> Le distribuzioni su istanze di Windows Server non supportano l'elemento. `runas` Se si esegue la distribuzione su istanze di Windows Server, non includerlo nel file. AppSpec

```
version: 0.0
os: linux
files:
  - source: Config/config.txt
    destination: /webapps/Config
  - source: source
    destination: /webapps/myApp
hooks:
  BeforeInstall:
    - location: Scripts/UnzipResourceBundle.sh
    - location: Scripts/UnzipDataBundle.sh
  AfterInstall:
    - location: Scripts/RunResourceTests.sh
      timeout: 180
  ApplicationStart:
    - location: Scripts/RunFunctionalTests.sh
      timeout: 3600
  ValidateService:
    - location: Scripts/MonitorService.sh
      timeout: 3600
      runas: codedeployuser
```

Per un'istanza di Windows Server, passa `os: linux` a. `os: windows` Inoltre, devi specificare i percorsi completi per `destination` (ad esempio, `c:\temp\webapps\Config` e `c:\temp\webapps\myApp`). Non includere l'elemento `runas`.

Ecco è la sequenza degli eventi durante la distribuzione:

1. Esegui lo script presente in `Scripts/UnzipResourceBundle.sh`.
2. Se lo script precedente ha restituito il codice di uscita 0 (operazione completata), esegui lo script presente in `Scripts/UnzipDataBundle.sh`.
3. Copia il file dal percorso `Config/config.txt` nel percorso `/webapps/Config/config.txt`.
4. Copia in modo ricorsivo tutti i file nella directory `source` nella directory `/webapps/myApp`.
5. Esegui lo script che si trova in `Scripts/RunResourceTests.sh` con un timeout di 180 secondi (3 minuti).
6. Esegui lo script che si trova in `Scripts/RunFunctionalTests.sh` con un timeout di 3600 secondi (1 ora).
7. Esegui lo script che si trova in `Scripts/MonitorService.sh` come utente `codedeploy` con un timeout di 3600 secondi (1 ora).

AppSpec Spaziatura tra i file

Di seguito è riportato il formato corretto per la spaziatura tra i AppSpec file. I numeri tra parentesi quadre indicano il numero di spazi che devono essere presenti tra gli elementi. Ad esempio, `[4]` significa inserire quattro spazi tra gli elementi. CodeDeploy genera un errore di cui potrebbe essere difficile eseguire il debug se le posizioni e il numero di spazi in un AppSpec file non sono corretti.

```
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
```

Ecco un esempio di file con AppSpec spaziatura corretta:

```
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
```

Per ulteriori informazioni sulla spaziatura, consulta la specifica YAML.

Convalida il AppSpec file e la posizione del file

Sintassi del file

Puoi utilizzare lo script AppSpec Assistant AWS fornito per convalidare il contenuto di un file. AppSpec È possibile trovare lo script insieme ai modelli di AppSpec file su. GitHub

Puoi anche utilizzare uno strumento basato su browser come YAML lint o il parser YAML online per aiutarti a controllare la sintassi YAML.

Posizione del file

Per verificare di aver inserito il AppSpec file nella directory principale della struttura di directory del contenuto di origine dell'applicazione, esegui uno dei seguenti comandi:

Nelle istanze locali di Linux, macOS o Unix:

```
ls path/to/root/directory/appspec.yml
```

Se il AppSpec file non si trova lì, viene visualizzato l'errore «Nessun file o directory di questo tipo».

Sulle istanze Windows locali:

```
dir path\to\root\directory\appspec.yml
```

Se il AppSpec file non si trova lì, viene visualizzato l'errore «File non trovato».

CodeDeploy riferimento alla configurazione dell'agente

Quando l' CodeDeploy agente viene installato, sull'istanza viene inserito un file di configurazione. Questo file di configurazione specifica i percorsi delle directory e altre impostazioni CodeDeploy da utilizzare durante l'interazione con l'istanza. Puoi modificare alcune delle opzioni di configurazione nel file.

Per le istanze Amazon Linux, Ubuntu Server e Red Hat Enterprise Linux (RHEL), il file di configurazione è denominato. `codedeployagent.yml` È collocato nella directory `/etc/codedeploy-agent/conf`.

Per le istanze di Windows Server, il file di configurazione è denominato. `conf.yml` È collocato nella directory `C:\ProgramData\Amazon\CodeDeploy`.

Le impostazioni di configurazione includono:

:log_aws_wire:	Imposta questa **`true`** opzione per consentir e all' CodeDeploy agente di acquisire i log dei cavi da Amazon S3 e di scriverli in un file di registro nella posizione indicata dall'impo stazione:log_dir:. Per la versione 2.0.x e successive, il nome del file è (creato con autorizzazioni 0640). `codedeploy-agent.aws_wire.log` Per la versione 1.8.x e precedenti, il nome del file è. `codedeploy-agent.wire.log` **Warning** È necessario impostare :log_aws_wire: a `true` solo per la quantità di tempo necessaria per acquisire log in rete. Il file wire log può raggiungere rapidamen te dimensioni molto grandi. L'output del wire log potrebbe contenere informazi oni riservate, incluso il contenuto in testo semplice dei file trasferiti da o verso Amazon S3 mentre questa impostazione era impostata su. `true` I wire log contengono informazioni su tutte le attività di Amazon S3 associate AWS all'account su cui era impostata questa impostazione, non solo sulle attività relative `true` CodeDeploy alle distribuzioni. L'impostazione predefinita è `false`.

	Questa impostazione vale per tutti i tipi di istanza. È necessario aggiungere questa impostazione di configurazione alle istanze di Windows Server per poterla utilizzare.
:log_dir:	La cartella sull'istanza in cui sono archiviati i file di registro relativi alle operazioni CodeDeploy dell'agente. L'impostazione predefinita è `'/var/log/aws/codedeploy-agent'` per le istanze Amazon Linux, Ubuntu Server e RHEL e `C:\ProgramData\Amazon\CodeDeploy\log` per le istanze di Windows Server.
:pid_dir:	La cartella in cui viene archiviato `codedeploy-agent.pid` . Questo file contiene l'ID di processo (PID) dell'agente. CodeDeploy L'impostazione predefinita è `'/opt/codedeploy-agent/state/.pid'` . Questa impostazione si applica solo alle istanze Amazon Linux, Ubuntu Server e RHEL.
:program_name:	Il nome del programma CodeDeploy dell'agente. L'impostazione predefinita è `codedeploy-agent`. Questa impostazione si applica solo alle istanze Amazon Linux, Ubuntu Server e RHEL.

:root_dir:	La cartella in cui vengono archiviate le revisioni correlate, la cronologia di distribuzione e gli script di distribuzione sull'istanza. L'impostazione predefinita è `/opt/codedeploy-agent/deployment-root` per le istanze Amazon Linux, Ubuntu Server e RHEL e `C:\ProgramData\Amazon\CodeDeploy` per le istanze di Windows Server.
:verbose:	Impostata `true` per consentire all' CodeDeploy agente di stampare i file di registro dei messaggi di debug sull'istanza. L'impostazione predefinita è `false`.
:wait_between_runs:	L'intervallo, in secondi, tra il polling CodeDeploy dell'agente CodeDeploy per le distribuzioni in sospeso. Per la versione 2.0.x e successive, l'impostazione predefinita è. `30` Per la versione 1.8.x e precedenti, l'impostazione predefinita è. `1`

:on_premises_config_file:	Per le istanze locali, il percorso verso una posizione alternativa per il file di configurazione denominato `codedeploy.onpremises.yml` (per Ubuntu Server e RHEL) o `conf.onpremises.yml` (per Windows Server). Per impostazione predefinita, questi file sono archiviati in `/etc/codedeploy-agent/conf/codedeploy.onpremises.yml` per Ubuntu Server e RHEL e `C:\ProgramData\Amazon\CodeDeploy\` per Windows Server. `conf.onpremises.yml` Polling dell'agente disponibile nella versione 1.0.1.686 e versioni successive dell'agente CodeDeploy.
:proxy_uri:	(Facoltativo) Il proxy HTTP tramite il quale si desidera che l' CodeDeploy agente si connetta AWS per le CodeDeploy operazioni. Utilizza un formato simile a `https://user:password@my.proxy:443/path?query` . Polling dell'agente disponibile nella versione 1.0.1.824 e versioni successive dell'agente CodeDeploy.

:max_revisions:	(Facoltativo) Il numero di revisioni dell'appl icazione per un gruppo di distribuzione che si desidera che l' CodeDeploy agente archivi. Qualsiasi revisione che supera il numero specificato viene eliminata. Inserire qualsiasi numero intero positivo. Se non viene specificato alcun valore, CodeDeploy manterrà le cinque revisioni più recenti oltre alla revisione attualmente distribuita. Supportate nella versione 1.0.1.966 e versioni successive dell'agente CodeDeploy.
Se non viene specificato alcun valore, manterrà le cinque revisioni più recenti oltre alla revisione attualmente distribuita. ----sep----:enable _auth_policy:	(Facoltativo) Imposta su `true` se desideri utilizzare l'autorizzazione IAM per configura re il controllo degli accessi e limitare l'autoriz zazione del ruolo o dell'utente IAM utilizzato dall'agente. CodeDeploy PerUtilizzo CodeDeplo y con Amazon Virtual Private Cloud, questo valore deve esseretrue. L'impostazione predefinita è `false`.
Per, questo valore deve essere. ----sep-- --:disable_imds_v1:	Questa impostazione è disponibile con l'agente 1.7.0 e versioni successive. CodeDeploy Impostata per `true` disabilitare il fallback su IMDSv1 quando si verificano errori IMDSv2. L'impostazione predefinita è (abilita il fallback). `false`
Il valore predefinito è (abilita il fallback). ---- sep----:disable_core_dumps:	Per la versione 2.0.x e successive. Impostato su per `true` sopprimere i core dump. Impostato `false` per le indagini sull'incidente. L'impostazione predefinita è `true`.

Impostato su per l'indagine sull'incidente. ----sep----:enable_command_port:	Per la versione 2.0.x e successive. Impostato per `true` abilitare l'interfaccia di gestione locale solo per il loopback (127.0.0.1) per interrogare lo stato dell'agente e iniettare i comandi localmente. L'impostazione predefinita è `false`.
Impostato per abilitare l'interfaccia di gestione locale solo per il loopback (127.0.0.1) per interrogare lo stato dell'agente e iniettare i comandi localmente. ----sep----:enable _archive_reuse:	Per la versione 2.1.0 e successive. Imposta `false` per scaricare sempre la revisione dell'applicazione, anche in una `RESTART` distribuzione, in cui la revisione è già presente sull'istanza. Quando questa impostazi one è impostata`true`, l'agente installa dalla copia già presente sull'istanza e scarica la revisione se tale copia è mancante o incomplet a. L'impostazione predefinita è `true`. Per ulteriori informazioni su questo comportam ento, consulta. Cronologia delle versioni dell'agente CodeDeploy
Per ulteriori informazioni su questo comportam ento, vedere. ----sep----:reject_symlinks _in_bundle:	Per la versione 2.0.x e successive. Imposta su per `true` rifiutare i pacchetti che contengono collegamenti simbolici o collegame nti fisici. L'impostazione predefinita è `false`.

Impostato su per rifiutare i pacchetti che contengono collegamenti simbolici o collegamenti fisici. ----sep----:reject_path_traversal_in_bundle:	Per la versione 2.0.x e successive. Imposta su `true` per rifiutare le voci di archivio che contengono `../` percorsi trasversali o assoluti e i campi di AppSpec origine che sfuggono alla radice del pacchetto. L'impostazione predefinita è `false`.
Impostato su per rifiutare le voci di archivio che contengono percorsi trasversali o assoluti e i campi di origine che sfuggono alla radice del pacchetto. ----sep----:reject_unsafe_permissions_in_bundle:	Per la versione 2.0.x e successive. Imposta su per `true` rifiutare i pacchetti che contengono SUID/SGID i file o le modalità di AppSpec autorizzazione impostate. SUID/SGID L'impostazione predefinita è `false`.
Imposta su per rifiutare i pacchetti che contengono file o le modalità di autorizzazione impostate. ----sep----:reject_unconfined_selinux_in_bundle:	Per la versione 2.0.x e successive. Imposta su per `true` rifiutare i pacchetti che contengono file con tipi SELinux. MAC-disabling L'impostazione predefinita è `false`.
Impostato su per rifiutare i pacchetti che contengono file con tipi SELinux. ----sep----:reject_symlink_permission_targets:	Per la versione 2.0.x e successive. Imposta su per `true` rifiutare l'`permissions` immissione di un AppSpec file la cui destinazione è un collegamento simbolico. Per impostazione predefinita, l'agente segue il collegamento quando applica modifiche alla proprietà, alla modalità, all'ACL e a SELinux. L'impostazione predefinita è `false`.

Per impostazione predefinita, l'agente segue il link quando applica le modifiche di proprietà , modalità, ACL e SELinux. ----sep----:ignore _ownership_in_bundle:	Per la versione 2.0.x e successive. `true`Impostare questa opzione per ripristin are la proprietà dei file estratti `tar` e dei `tgz` pacchetti `root` anziché applicare l'ID utente e l'ID di gruppo registrati nell'archivio. I file contenuti nei `zip` pacchetti sono di proprietà di `root` indipendentemente da questa impostazi one. L'impostazione predefinita è `false`.
I file nei bundle sono di proprietà di indipende ntemente da questa impostazione. ----sep-- --:restrict_agent_dir_permissions:	Per la versione 2.0.x e successive. Solo istanze Linux. `true`Impostare questa opzione per limitare l'accesso alle directory e ai file presenti nella directory principale di installazione dell'agen te all'accesso solo come root anziché alle modalità leggibili in tutto il mondo utilizzate per impostazione predefinita. L'impostazione predefinita è `false`.
Impostato per limitare le directory e i file nella radice di installazione dell'agente all'accesso solo come root invece delle modalità leggibili da tutto il mondo utilizzate di default. ----sep-- --:restrict_log_dir_permissions:	Per la versione 2.0.x e successive. Solo istanze Linux. `true`Impostare questa opzione per limitare l'accesso alla directory del registro dell'agente, che contiene il registro dell'agente e il registro dell'aggiornamento, all'accesso solo come root anziché alle modalità leggibili in tutto il mondo utilizzate per impostazione predefinita. L'impostazione predefinita è `false`.

Impostato per limitare la directory del registro dell'agente, che contiene il registro dell'agen te e il registro dell'aggiornamento, all'accesso solo come root anziché alle modalità leggibili dal mondo utilizzate per impostazione predefini ta. ----sep----:restrict_hook_env_to_allowlist:	Per la versione 2.0.x e successive. Imposta su per `true` eseguire gli hook degli eventi del ciclo di vita con un ambiente minimo che contenga solo le variabili di shell di base e le variabili di distribuzione. CodeDeploy Per impostazione predefinita, gli hook ereditano l'intero ambiente del processo agente. L'impostazione predefinita è `false`.
Per impostazione predefinita, gli hook ereditano l'intero ambiente del processo agente. ----sep-- --:strip_loader_env_in_hooks:	Per la versione 2.0.x e successive. Istanze Linux e istanze macOS con versione 2.1.2 e successive. Imposta su `true` per rimuovere le `LD_AUDIT` variabili `LD_PRELOAD` `LD_LIBRARY_PATH` , e dall'ambiente degli hook degli eventi del ciclo di vita. Su macOS, l'impostazione rimuove invece le `DYLD_VERSIONED_FRAMEWORK_PA TH` variabili `DYLD_INSERT_LIBRAR IES` `DYLD_LIBRARY_PATH` `DYLD_FRAM EWORK_PATH` ,`DYLD_FALLBACK_LIBR ARY_PATH` ,`DYLD_FALLBACK_FRAM EWORK_PATH` ,`DYLD_VERSIONED_LIB RARY_PATH` , e. L'impostazione predefinita è `false`.

Su macOS, l'impostazione rimuove invece le variabili,,,, e. ----sep----:disable_powersh ell_profile_in_hooks:	Per la versione 2.0.x e successive. Solo istanze di Windows Server. Imposta su per `true` eseguire gli hook degli eventi `.ps1` del ciclo di vita con le opzioni and. `-NoProfile -NonInteractive` Per impostazione predefinita, questi hook caricano il PowerShell profilo dell'utente con cui viene eseguito l'agente. L'impostazione predefinita è `false`.
Per impostazione predefinita, questi hook caricano il profilo dell'utente con cui viene eseguito l'agente. ----sep----:archive_max_ext raction_size:	Per la versione 2.0.x e successive. Imposta un limite alla dimensione dichiarata del pacchetto non compresso. Se il contenuto estratto supera questo valore, la distribuzione non riesce. L'impostazione predefinita è no cap (unlimited).
L'impostazione predefinita è no cap (unlimited). ----sep----:wait_after_error:	Il numero di secondi che l'agente attende dopo aver riscontrato un errore prima di riprendere il normale funzionamento. L'impostazione predefinita è `30`.
Il numero di secondi che l'agente attende dopo aver riscontrato un errore prima di riprender e il normale funzionamento. ----sep----:http_r ead_timeout:	Il timeout di lettura HTTP in secondi per le chiamate API di servizio. L'impostazione predefinita è `80`.
Il timeout di lettura HTTP in secondi per le chiamate all'API di servizio. ----sep----:kill_a gent_max_wait_time_seconds:	Il numero massimo di secondi in cui l'agente attende un arresto regolare prima della chiusura forzata. Per la versione 2.0.x e successive, questo valore è limitato a 24 ore (86400 secondi). L'impostazione predefinita è `7200` (2 ore).

L'impostazione predefinita è (2 ore). ----sep----:enable_deployments_log:	Impostato per scrivere il log delle distribuzioni aggregate (). `true` `codedeploy-agent-deployments.log` L'impostazione predefinita è `true`.
Impostato su per scrivere il log delle distribuzioni aggregate (). ----sep----:use_fips_mode:	Impostato per utilizzare gli endpoint. `true` FIPS-compliant Si applica solo alle regioni degli Stati Uniti. L'impostazione predefinita è `false`.
Si applica solo alle regioni degli Stati Uniti. ----sep----:deploy_control_endpoint:	(Facoltativo) Un URL dell'endpoint del piano di controllo personalizzato per il servizio. CodeDeploy L'impostazione predefinita è nessun override (l'agente risolve l'endpoint dalla regione).
L'impostazione predefinita è nessun override (l'agente risolve l'endpoint dalla Regione). ----sep----:s3_endpoint_override:	(Facoltativo) Un URL endpoint Amazon S3 personalizzato per i download in bundle. L'impostazione predefinita non prevede l'override (l'agente risolve l'endpoint Amazon S3 dalla regione).

L'impostazione predefinita è nessun override (l'agente risolve l'endpoint Amazon S3 dalla regione). ----sep----:s3_download_max _concurrency:	Per la versione 2.1.2 e successive. Il numero di richieste di intervallo HTTP parallele utilizzate dall'agente per scaricare una revisione dell'applicazione da Amazon S3. Imposta su un valore maggiore rispetto 1 a quello per scaricare revisioni di grandi dimensioni in intervalli paralleli. Con1, l'agente scarica la revisione con una sola `GetObject` richiesta. I valori di cui sopra 64 vengono trattati come 64 e `0` vengono trattati come1. L'impostazione predefinita è 1.
I valori sopra indicati vengono trattati come e vengono trattati come. ----sep----:s3_dow nload_chunk_size:	Per la versione 2.1.2 e successive. La dimensione di ogni intervallo richiesto quando `:s3_download_max_concurrenc y:` è maggiore di1. Immettete un numero di byte o una stringa di dimensioni come `"16MiB"` o`"16MB"`. Una revisione di queste dimensioni o inferiori viene scaricata con una sola richiesta. I valori inferiori a 64 KiB vengono trattati come 64 KiB. L'impostazione predefinita è `"8MiB"`.
I valori inferiori a 64 KiB vengono trattati come 64 KiB. ----sep----:ongoing_deployment_track ing:	Il nome della sottodirectory utilizzata per il monitoraggio dello stato della distribuzione. L'impostazione predefinita è `ongoing-d eployment` .

Argomenti correlati

Collaborazione con l' CodeDeploy agente

Gestione delle operazioni degli CodeDeploy agenti

CloudFormation modelli di CodeDeploy riferimento

Questa sezione presenta CloudFormation le risorse, la trasformazione e l'hook progettati per funzionare con le distribuzioni. CodeDeploy Per una procedura dettagliata sulla creazione di un aggiornamento dello stack gestito dall'hook for, vedi CloudFormation CodeDeploy Crea una blue/green distribuzione Amazon ECS tramite CloudFormation

> **Note**
>
> CloudFormation gli hook fanno parte dei CloudFormation componenti AWS e sono diversi dagli CodeDeploy hook degli eventi del ciclo di vita.

Oltre agli altri metodi disponibili CodeDeploy, puoi utilizzare i CloudFormation modelli per eseguire le seguenti attività:

- Creare applicazioni
- Creare gruppi di distribuzione e specificare una revisione di destinazione.
- Creare configurazioni della distribuzione
- Crea istanze Amazon EC2.

CloudFormation è un servizio che ti aiuta a modellare e configurare AWS le tue risorse utilizzando modelli. Un CloudFormation modello è un file di testo il cui formato è conforme allo standard JSON. Crei un modello che descrive tutte le AWS risorse che desideri e si CloudFormation occupa del provisioning e della configurazione di tali risorse per te.

Per ulteriori informazioni, consulta Cos'è? AWS CloudFormation e Utilizzo dei AWS CloudFormation modelli nella Guida AWS CloudFormation per l'utente.

Se si prevede di utilizzare CloudFormation modelli compatibili con la propria organizzazione, CodeDeploy in qualità di amministratore è necessario concedere l'accesso ai CloudFormation AWS servizi e alle azioni CloudFormation da cui dipende. Per concedere le autorizzazioni per creare

applicazioni, gruppi di distribuzione e configurazioni di distribuzione, aggiungi la seguente politica al set di autorizzazioni degli utenti con cui lavoreranno: CloudFormation

JSON

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

Per ulteriori informazioni sulle politiche, consulta i seguenti argomenti:

- Per visualizzare la policy da aggiungere al set di autorizzazioni degli utenti che creeranno istanze Amazon EC2, consulta. Crea un'istanza Amazon EC2 per (CodeDeploy CloudFormation modello)
- Per informazioni sull'aggiunta di policy ai set di autorizzazioni, consulta Create a permission set nella IAM User Guide.
- Per informazioni su come limitare gli utenti a un insieme limitato di CodeDeploy azioni e risorse, consultaAWS politiche gestite (predefinite) per CodeDeploy.

La tabella seguente mostra le azioni che un CloudFormation modello può eseguire per conto dell'utente e include collegamenti a ulteriori informazioni sui tipi di AWS risorse e sui relativi tipi di proprietà che è possibile aggiungere a un CloudFormation modello.

Azione	CloudFormation riferimento	tipo di riferimento
Crea un' CodeDeploy applicazione.	AWS::CodeDeploy: :applicazione	CloudFormation risorsa
Crea e specifica i dettagli per un gruppo di distribuzione	AWS::CodeDeploy::DeploymentGroup	CloudFormation risorsa

Azione	CloudFormation riferimento	tipo di riferimento
da utilizzare per distribuire le revisioni delle applicazioni. [1]		
Crea un set di regole di distribuzione, condizioni di successo dell'implementazione e condizioni di errore di CodeDeploy distribuzione da utilizzare durante una distribuzione.	AWS::CodeDeploy::DeploymentConfig	CloudFormation risorsa
Crea un'istanza Amazon EC2. [2]	AWS: :EC2: :istanza	CloudFormation risorsa
Usa CloudFormation `AWS::CodeDeployBlueGreen` transform and `AWS::CodeDeploy::BlueGreen` hook per gestire gli aggiornamenti dello stack, creare risorse e spostare il traffico per le CodeDeploy blue/green implementazioni. [3]	AWS::CodeDeployBlueGreen	La trasformazione `AWS::CodeDeployBlueGreen` è una macro ospitata da CloudFormation
	AWS::CodeDeploy::BlueGreen	L'`AWS::CodeDeploy::BlueGreen` hook è strutturato come una `Hook` risorsa in CloudFormation. L'hook include parametri che sostituiscono il CodeDeploy AppSpec file puntando agli hook di eventi del CodeDeploy ciclo di vita designati.

Azione	CloudFormation riferimento	tipo di riferimento
[1] Se si specifica la versione della revisione dell'applicazione che si desidera distribuire come parte del gruppo di distribuzione, la revisione di destinazione verrà distribuita non appena il processo di provisioning sarà completo. Per ulteriori informazioni sulla configurazione del modello, consulta la revisione di distribuzione S3Location e la revisione della CodeDeploy DeploymentGroup distribuzione nella Guida per l'utente. CodeDeploy Deploymen tGroup GitHubLocation AWS CloudFormation [2] Forniamo modelli che puoi utilizzare per creare istanze Amazon EC2 nelle regioni in cui CodeDeploy è supportata. Per ulteriori informazioni sull'utilizzo di questi modelli, consulta Crea un'istanza Amazon EC2 per (CodeDeploy CloudFormation modello). [3] Solo le blue/green distribuzioni di Amazon ECS sono supportat e da questa configurazione di distribuzione. Per ulteriori informazioni sulle configurazioni di distribuzione per le blue/gree n distribuzioni di Amazon ECS tramite, consulta. CloudForm ationConfigurazioni di distribuzione per CloudFormation blue/ green distribuzioni (Amazon ECS) Per ulteriori informazioni sulle blue/green distribuzioni di Amazon ECS CloudFormation e su come visualizzarle in CodeDeploy, consulta. Crea una blue/gree n distribuzione Amazon ECS tramite CloudFormation		

Utilizzo CodeDeploy con Amazon Virtual Private Cloud

Se utilizzi Amazon Virtual Private Cloud (Amazon VPC) per ospitare AWS le tue risorse, puoi stabilire una connessione privata tra il tuo VPC e. CodeDeploy Puoi utilizzare questa connessione CodeDeploy per consentire di comunicare con le tue risorse sul tuo VPC senza passare attraverso la rete Internet pubblica.

Amazon VPC è un AWS servizio che puoi utilizzare per avviare AWS risorse in una rete virtuale definita dall'utente. Con un VPC;, detieni il controllo delle impostazioni della rete, come l'intervallo di indirizzi IP, le sottoreti, le tabelle di routing e i gateway di rete. Con gli endpoint VPC, il routing

tra il VPC e i AWS servizi viene gestito dalla AWS rete e puoi utilizzare le policy IAM per controllare l'accesso alle risorse del servizio.

Per connettere il tuo VPC CodeDeploy, definisci un'interfaccia VPC endpoint per. CodeDeploy Un endpoint di interfaccia è un'interfaccia di rete elastica con un indirizzo IP privato che funge da punto di ingresso per il traffico destinato a un servizio supportato AWS. L'endpoint fornisce una connettività affidabile e scalabile CodeDeploy senza richiedere un gateway Internet, un'istanza NAT (Network Address Translation) o una connessione VPN. Per ulteriori informazioni, consulta Che cos'è Amazon VPC? nella Guida per l'utente Amazon VPC.

Gli endpoint VPC di interfaccia sono alimentati da AWS PrivateLink, una AWS tecnologia che consente la comunicazione privata tra AWS i servizi utilizzando un'interfaccia di rete elastica con indirizzi IP privati. Per ulteriori informazioni, consulta AWS PrivateLink.

Le fasi seguenti sono per gli utenti Amazon VPC. Per ulteriori informazioni, consulta l'argomento relativo alle nozioni di base nella Guida per l'utente di Amazon VPC.

Disponibilità

CodeDeploy ha due endpoint VPC: uno per le operazioni degli CodeDeploy agenti e uno per CodeDeploy le operazioni API. La tabella seguente mostra le AWS regioni supportate per ogni endpoint.

Per ulteriori informazioni e per gli endpoint FIPS, consulta AWS CodeDeploy endpoint e quote.

Nome della Regione	Codice regione	Endpoint dell'agente	Endpoint API	Regione conforme a FIPS?
Stati Uniti orientali (Virginia settentrionale)	us-east-1	Sì	Sì	Sì
Stati Uniti orientali (Ohio)	us-east-2	Sì	Sì	Sì
Stati Uniti occidentali	us-west-1	Sì	Sì	Sì

Nome della Regione	Codice regione	Endpoint dell'agente	Endpoint API	Regione conforme a FIPS?
(California settentrionale)				
Stati Uniti occidentali (Oregon)	us-west-2	Sì	Sì	Sì
Africa (Città del Capo)	af-south-1	Sì	Sì	
Asia Pacific (Hong Kong)	ap-east-1	Sì	Sì	
Asia Pacifico (Hyderabad)	ap-south-2	Sì	Sì	
Asia Pacifico (Giacarta)	ap-southeast-3	Sì	Sì	
Asia Pacifico (Melbourne)	ap-southeast-4	Sì	Sì	
Asia Pacifico (Malesia)	ap-southeast-5	Sì	Sì	
Asia Pacifico (Nuova Zelanda)	ap-southeast-6	Sì	Sì	
Asia Pacifico (Thailandia)	ap-southeast-7	Sì	Sì	
Asia Pacifico (Taipei)	ap-east-2	Sì	Sì	
Asia Pacifico (Mumbai)	ap-south-1	Sì	Sì	

Nome della Regione	Codice regione	Endpoint dell'agente	Endpoint API	Regione conforme a FIPS?
Asia Pacifico (Osaka)	ap-northeast-3	Sì	Sì	
Asia Pacifico (Seoul)	ap-northeast-2	Sì	Sì	
Asia Pacifico (Singapore)	ap-southeast-1	Sì	Sì	
Asia Pacifico (Sydney)	ap-southeast-2	Sì	Sì	
Asia Pacifico (Tokyo)	ap-northeast-1	Sì	Sì	
Canada (Centrale)	ca-central-1	Sì	Sì	
Cina (Pechino)	cn-north-1	Sì	No	
China (Ningxia)	cn-northwest-1	No	No	
Europa (Francoforte)	eu-central-1	Sì	Sì	
Europa (Irlanda)	eu-west-1	Sì	Sì	
Europa (Londra)	eu-west-2	Sì	Sì	
Europe (Milan)	eu-south-1	Sì	Sì	
Europa (Parigi)	eu-west-3	Sì	Sì	
Europa (Spagna)	eu-south-2	Sì	Sì	

Nome della Regione	Codice regione	Endpoint dell'agente	Endpoint API	Regione conforme a FIPS?
Europa (Stoccolma)	eu-north-1	Sì	Sì	
Europa (Zurigo)	eu-central-2	Sì	Sì	
Israele (Tel Aviv)	il-central-1	Sì	Sì	
Medio Oriente (Bahrein)	me-south-1	Sì	Sì	
Medio Oriente (Emirati Arabi Uniti)	me-central-1	Sì	Sì	
Messico (Centrale)	mx-central-1	Sì	Sì	
Sud America (San Paolo)	sa-east-1	Sì	Sì	
AWS GovCloud (US-East)	us-gov-east-1	No	No	Sì
AWS GovCloud (US-West)	us-gov-west-1	No	No	Sì

Crea endpoint VPC per CodeDeploy

Per iniziare a utilizzarlo CodeDeploy con il tuo VPC, crea un endpoint VPC di interfaccia per. CodeDeploy CodeDeployrichiede endpoint separati per le operazioni dell'agente Git e per le operazioni CodeDeploy API. A seconda delle esigenze aziendali, potrebbe essere necessario creare più endpoint VPC. Quando crei un endpoint VPC per CodeDeploy, scegli AWS Servizi e in Nome servizio scegli una delle seguenti opzioni:

- com.amazonaws. ***region***.codedeploy: scegli questa opzione se desideri creare un endpoint VPC per le operazioni API. CodeDeploy Ad esempio, scegli questa opzione se gli utenti utilizzano l'AWS CLI CodeDeploy API o gli AWS SDK con cui interagire CodeDeploy per operazioni come, e. `CreateApplication` `GetDeployment` `ListDeploymentGroups`
- com.amazonaws. ***region***.codedeploy-fips: scegli questa opzione se desideri creare un endpoint VPC per le operazioni API per l'endpoint FIPS. CodeDeploy
- com.amazonaws. ***region***.codedeploy-commands-secure: scegli questa opzione se desideri creare un endpoint VPC per le operazioni degli agenti. CodeDeploy Dovrai inoltre impostarlo nel file di configurazione dell'agente e `:enable_auth_policy:` allegare le `true` autorizzazioni richieste. Per ulteriori informazioni, consulta Configura l'agente e le autorizzazioni IAM CodeDeploy.

Se utilizzi implementazioni Lambda o ECS, devi solo creare un endpoint VPC per com.amazonaws. ***region***.codedeploy. I clienti che utilizzano implementazioni Amazon EC2 avranno bisogno di endpoint VPC per entrambi i sistemi com.amazonaws. *region*.codedeploy e com.amazonaws. ***region***.codedeploy-commands-secure.

Configura l'agente e le autorizzazioni IAM CodeDeploy

Per utilizzare gli endpoint Amazon VPC con CodeDeploy, devi impostare il valore di `:enable_auth_policy:` to `true` nel file di configurazione dell'agente che si trova sulle tue istanze EC2 o locali. Per ulteriori informazioni sul file di configurazione dell'agente, consulta. CodeDeploy riferimento alla configurazione dell'agente

È inoltre necessario aggiungere le seguenti autorizzazioni IAM al profilo dell'istanza Amazon EC2 (se si utilizzano istanze Amazon EC2) o all'utente o al ruolo IAM (se si utilizzano istanze locali).

```
{
  "Statement": [
    {
      "Action": [
        "codedeploy-commands-secure:GetDeploymentSpecification",
        "codedeploy-commands-secure:PollHostCommand",
        "codedeploy-commands-secure:PutHostCommandAcknowledgement",
        "codedeploy-commands-secure:PutHostCommandComplete"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
```

```
}
```

Per ulteriori informazioni, consulta Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC.

CodeDeploy riferimento al kit di risorse

Molti dei file su cui CodeDeploy si basa sono archiviati in bucket AWS Region-specific Amazon S3 disponibili pubblicamente. Questi file includono file di installazione per l' CodeDeploy agente, modelli e file di applicazione di esempio. Questa raccolta di file è denominata CodeDeploy Resource Kit.

Argomenti

- Nomi dei bucket del Resource Kit per regione
- Contenuto del kit di risorse
- Visualizza un elenco dei file del Resource Kit
- Scarica i file del kit di risorse

Nomi dei bucket del Resource Kit per regione

Questa tabella elenca i nomi delle *bucket-name* sostituzioni necessarie per alcune procedure della guida. Questi sono i nomi dei bucket Amazon S3 che contengono i file CodeDeploy Resource Kit.

> **Note**
>
> Per accedere al bucket Amazon S3 nella regione Asia Pacifico (Hong Kong), devi abilitare la regione nel tuo account. AWS Per ulteriori informazioni, consulta Gestione delle AWS regioni.

Nome della Regione	***Bucket-name*** sostituzione	Identificatore della regione
Stati Uniti orientali (Virginia settentrionale)	aws-codedeploy-us-east-1	us-east-1
Stati Uniti orientali (Ohio)	aws-codedeploy-us-east-2	us-east-2
Stati Uniti occidentali (California settentrionale)	aws-codedeploy-us-west-1	us-west-1

Nome della Regione	***Bucket-name*** sostituzione	Identificatore della regione
Stati Uniti occidentali (Oregon)	aws-codedeploy-us-west-2	us-west-2
Africa (Città del Capo)	aws-codedeploy-af-sud-1	af-south-1
Asia Pacifico (Hong Kong)	aws-codedeploy-ap-east-1	ap-east-1
Asia Pacifico (Hyderabad)	aws-codedeploy-ap-sud-2	ap-south-2
Asia Pacifico (Giacarta)	aws-codedeploy-ap-southeast-3	ap-southeast-3
Asia Pacifico (Melbourne)	aws-codedeploy-ap-southeast-4	ap-southeast-4
Asia Pacifico (Malesia)	aws-codedeploy-ap-southeast-5	ap-southeast-5
Asia Pacifico (Nuova Zelanda)	aws-codedeploy-ap-southeast-6	ap-southeast-6
Asia Pacifico (Thailandia)	aws-codedeploy-ap-southeast-7	ap-southeast-7
Asia Pacifico (Taipei)	aws-codedeploy-ap-east-2	ap-east-2
Asia Pacifico (Mumbai)	aws-codedeploy-ap-south-1	ap-south-1
Asia Pacifico (Osaka)	aws-codedeploy-ap-northeast-3	ap-northeast-3
Asia Pacifico (Seul)	aws-codedeploy-ap-northeast-2	ap-northeast-2
Asia Pacifico (Singapore)	aws-codedeploy-ap-southeast-1	ap-southeast-1
Asia Pacifico (Sydney)	aws-codedeploy-ap-southeast-2	ap-southeast-2

Nome della Regione	***Bucket-name*** sostituzione	Identificatore della regione
Asia Pacifico (Tokyo)	aws-codedeploy-ap-northeast-1	ap-northeast-1
Canada (Centrale)	aws-codedeploy-it-central-1	ca-central-1
Europa (Francoforte)	aws-codedeploy-eu-central-1	eu-central-1
Europa (Irlanda)	aws-codedeploy-eu-west-1	eu-west-1
Europa (Londra)	aws-codedeploy-eu-west-2	eu-west-2
Europa (Milano)	aws-codedeploy-eu-sud-1	eu-south-1
Europa (Parigi)	aws-codedeploy-eu-west-3	eu-west-3
Europa (Spagna)	aws-codedeploy-eu-south-2	eu-south-2
Europa (Stoccolma)	aws-codedeploy-eu-north-1	eu-north-1
Europa (Zurigo)	aws-codedeploy-eu-central-2	eu-central-2
Israele (Tel Aviv)	aws-codedeploy-il-central-1	il-central-1
Medio Oriente (Bahrein)	aws-codedeploy-me-sud-1	me-south-1
Medio Oriente (Emirati Arabi Uniti)	aws-codedeploy-me-central-1	me-central-1
Messico (centrale)	aws-codedeploy-mx-central-1	mx-central-1
Sud America (San Paolo)	aws-codedeploy-it-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

Contenuto del kit di risorse

La tabella seguente elenca i file del CodeDeploy Resource Kit.

File	Description
`LATEST_VERSION`	Un file utilizzato da meccanismi di aggiornamento come Amazon EC2 Systems Manager per determinare la versione più recente CodeDeploy dell'agente.
`VERSION`	Il meccanismo di aggiornamento automatico è stato rimosso nella versione 1.1.0 CodeDeploy dell'agente e questo file non viene più utilizzato. Un file utilizzato dagli agenti CodeDeploy per aggiornarne l'esecuzione sulle istanze.
`codedeploy-agent.noarch.rpm`	L' CodeDeploy agente per Amazon Linux e Red Hat Enterprise Linux (RHEL). Potrebbero esserci diversi file con lo stesso nome di file di base, ma versioni differenti (ad esempio `-1.0-0`).
`codedeploy-agent_all.deb`	L' CodeDeploy agente per Ubuntu Server. Potrebbero esserci diversi file con lo stesso nome di file di base, ma versioni differenti (ad esempio `_1.0-0`).
`codedeploy-agent.msi`	L' CodeDeploy agente per Windows Server. Potrebbero esserci diversi file con lo stesso nome di file di base, ma versioni differenti (ad esempio `-1.0-0`).
`install`	Un file che puoi utilizzare per installare più facilmente l' CodeDeploy agente. Il `latestv2/` prefisso serve allo script di installazione della versione 2.0.x.

File	Description
`CodeDeploy_SampleCF_Template.json`	Un AWS CloudFormation modello che puoi utilizzare per avviare da una a tre istanze Amazon EC2 che eseguono Amazon Linux o Windows Server. Potrebbero esserci diversi file con lo stesso nome di file di base, ma versioni differenti (ad esempio `-1.0.0`).
`CodeDeploy_SampleCF_ELB_Integration.json`	Un AWS CloudFormation modello che puoi utilizzare per creare un sito Web di esempio con bilanciamento del carico in esecuzione su un server Web Apache. L'applicazione è configurata per coprire tutte le zone di disponibilità nella regione in cui è stata creata. Questo modello crea tre istanze Amazon EC2 e un profilo di istanza IAM per concedere alle istanze l'accesso alle risorse in Amazon S3, Amazon EC2 Auto Scaling ed Elastic Load Balancing. CloudFormation Inoltre, crea un sistema di bilanciamento del carico e un ruolo di servizio. CodeDeploy
`SampleApp_ELB_Integration.zip`	Un esempio di revisione dell'applicazione che puoi distribuire su un'istanza Amazon EC2 registrata su un sistema di bilanciamento del carico Elastic Load Balancing.
`SampleApp_Linux.zip`	Un esempio di revisione dell'applicazione che puoi distribuire su un'istanza Amazon EC2 che esegue Amazon Linux o su un'istanza Ubuntu Server o RHEL. Potrebbero esserci diversi file con lo stesso nome di file di base, ma versioni differenti (ad esempio `-1.0`).

File	Description
`SampleApp_Windows.zip`	Un esempio di revisione dell'applicazione che puoi distribuire su un'istanza di Windows Server. Potrebbero esserci diversi file con lo stesso nome di file di base, ma versioni differenti (ad esempio `-1.0`).

Visualizza un elenco dei file del Resource Kit

Per visualizzare un elenco di file, usa il aws s3 ls comando relativo alla tua regione.

> **Note**
>
> I file in ogni bucket sono progettati per funzionare con le risorse della regione corrispondente.

- `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`

Scarica i file del kit di risorse

Per scaricare un file, usa il aws s3 cp comando relativo alla tua regione.

> **Note**
>
> Assicurati di utilizzare il punto (`.`) in prossimità della fine. Questo consente di scaricare il file nella directory corrente.

Ad esempio, i seguenti comandi scaricano un singolo file denominato `SampleApp_Linux.zip` da una delle cartelle del bucket `/samples/latest/`:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip . --region us-east-2`
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip . --region us-east-1`
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip . --region us-west-1`
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip . --region us-west-2`
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip . --region ca-central-1`
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip . --region eu-west-1`
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip . --region eu-west-2`
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip . --region eu-west-3`
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip . --region eu-central-1`
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Linux.zip . --region il-central-1`
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Linux.zip . --region ap-east-1`
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip . --region ap-northeast-1`

- ```
  aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip . --
  region ap-northeast-2
  ```
- ```
  aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip . --
  region ap-southeast-1
  ```
- ```
  aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip . --
  region ap-southeast-2
  ```
- ```
  aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Linux.zip . --
  region ap-southeast-4
  ```
- ```
  aws s3 cp s3://aws-codedeploy-ap-southeast-5/samples/latest/SampleApp_Linux.zip . --
  region ap-southeast-5
  ```
- ```
  aws s3 cp s3://aws-codedeploy-ap-southeast-6/samples/latest/SampleApp_Linux.zip . --
  region ap-southeast-6
  ```
- ```
  aws s3 cp s3://aws-codedeploy-ap-southeast-7/samples/latest/SampleApp_Linux.zip . --
  region ap-southeast-7
  ```
- ```
  aws s3 cp s3://aws-codedeploy-ap-east-2/samples/latest/SampleApp_Linux.zip . --region
   ap-east-2
  ```
- ```
  aws s3 cp s3://aws-codedeploy-mx-central-1/samples/latest/SampleApp_Linux.zip . --
  region mx-central-1
  ```
- ```
  aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip . --
  region ap-south-1
  ```
- ```
  aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip . --region
   sa-east-1
  ```

Per scaricare tutti i file, usa uno dei seguenti comandi per la tua regione:

- ```
  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 quote

Le tabelle seguenti descrivono le quote in. CodeDeploy

> **Note**
>
> Il limite EC2/On-Premises di ore di esecuzione della distribuzione sul posto varia. Per le configurazioni di distribuzione personalizzate create prima di giugno 2023, il limite è di 8 ore. Per le configurazioni di distribuzione personalizzate create a giugno 2023 o successivamente, il limite è di 12 ore. Per le configurazioni di distribuzione predefinite, il limite è di 12 ore.

Nome	Predefinita	Adatta e	Description
AWS L'implementazione Lambda può essere eseguita in poche ore	Ogni Regione supportata: 50	No	Numero massimo di ore di esecuzione di un'implementazione AWS Lambda (48 ore per il tempo massimo tra il primo e l'ultimo turno di traffico più un'ora per ciascuno dei due possibili collegamenti del ciclo di vita)
Applicazioni associate per account per regione	Ogni regione supportata: 1.000	Sì	Il numero massimo di applicazioni associate a un AWS account in una singola regione

Nome	Predefinita	Adatta e	Description
Allarmi associati per gruppo di distribuzione	Ogni Regione supportata: 50	Sì	Il numero massimo di allarmi associati a un gruppo di distribuzione
Gruppo Auto Scaling in un gruppo di implementazione	Ogni regione supportata: 10	Sì	Numero massimo di gruppi con dimensionamento automatico Amazon EC2 in un gruppo di implement azione
Implementazioni simultanee per account	Ogni regione supportata: 1.300	Sì	Numero massimo di distribuzioni simultanee associate a un account. AWS Ogni implement azione su un'istanza Amazon EC2 aumentata in un gruppo con dimensionamento automatico Amazon EC2 viene conteggiata come una singola implement azione simultanea per ogni applicazione a cui è associata l'istanza EC2.

Nome	Predefinita	Adatta e	Description
Implementazioni simultanee per gruppo di implementazione	Ogni regione supportata: 1	No	Numero massimo di implementazioni simultanee in un gruppo di implementazione. Questo limite ha lo scopo di evitare implementazioni simultanee della stessa applicazione sullo stesso gruppo di implement azione.
Configurazioni di implementazione personalizzate per account	Ogni Regione supportata: 50	No	Numero massimo di configurazioni di distribuz ione personalizzate associate a un account AWS
Gruppi di implementazione associati a una singola applicazione	Ogni regione supportata: 1.000	Sì	Numero massimo di gruppi di distribuzioni associati a un'unica applicazione
EC2/On-Premises blue/green l'install azione viene eseguita in poche ore	Ogni regione supportata: 109	No	Numero massimo di ore di esecuzione di un' EC2/ On-Premises blue/green implementazione (48 ore per ciascuno dei due limiti precedenti più un'ora per ciascuno dei 13 possibili eventi del ciclo di vita)

Nome	Predefinita	Adatta e	Description
EC2/On-Premises l'installazione sul posto viene eseguita in poche ore	Ogni regione supportata: 12	No	Numero massimo di ore di esecuzione di EC2/On-Premises una distribuzione sul posto
Trigger di notifica eventi in un gruppo di implementazione	Ogni regione supportata: 10	Sì	Numero massimo di trigger di notifica eventi in un gruppo di distribuzione
GitHub token di connessione per account	Ogni regione supportata: 25	No	Numero massimo di token di GitHub connessione per un singolo account AWS
Ore tra il completamento di una distribuzione e la chiusura delle istanze originali durante una distribuzione EC2/On-Premises blue/green	Ogni regione supportata: 48	No	Numero massimo di ore tra il completamento di una distribuzione e la chiusura delle istanze originali durante una distribuzione EC2/On-Premises blue/green
Ore che intercorrono tra l'implementazione di una revisione e il momento in cui il traffico si sposta verso le istanze sostitutive durante un'implementazione EC2/On-Premises blue/green	Ogni regione supportata: 48	No	Numero massimo di ore tra l'implementazione di una revisione e il momento in cui il traffico si sposta verso le istanze sostitutive durante un'implementazione EC2/On-Premises blue/green

Nome	Predefinita	Adatta e	Description
Conteggio delle istanze per implement azione	us-east-1: 2.000 us-east-2:2.000 us-west-2: 2.000 eu-central-1: 2.000 Tutte le altre regioni supportate: 1.000	Sì	Numero massimo di istanze in una singola distribuzione
Minuti che una blue/green distribuzione può attendere dopo una distribuzione riuscita prima di terminare le istanze dalla distribuzione originale	Ogni regione supportata: 2.800	No	Numero massimo di minuti che una blue/green distribuzione può attendere dopo una distribuzione riuscita prima di terminare le istanze dalla distribuzione originale
Minuti tra il primo e l'ultimo cambio di traffico durante una distribuzione AWS Lambda, canaria o lineare	Ogni regione supportata: 2.880	No	Numero massimo di minuti tra il primo e l'ultimo cambio di traffico durante una distribuzione AWS Lambda, canaria o lineare

Nome	Predefinita	Adatta e	Description
Numero di minuti prima che una implementazione abbia esito negativo se un evento del ciclo di vita non viene avviato	Ogni Regione supportata: 5	No	Numero massimo di minuti prima che una distribuzione fallisca se un evento del ciclo di vita non inizia dopo (1) l'attivazione di una distribuzione utilizzando la console o il comando CREATE-DEPLOYMENT della AWS CLI oppure (2) il completamento dell'evento del ciclo di vita precedente.
Numero di gruppi di implementazione che possono essere associati a un servizio Amazon ECS	Ogni regione supportata: 1	No	Numero massimo di gruppi di implement azione che possono essere associati a un servizio Amazon ECS
Numero di istanze che possono essere passate all'azione API BatchGetO nPremisesInstances	Ogni regione supportata: 100	No	Numero massimo di istanze che possono essere passate all'azione API BatchGetOnPremises Instances
Numero di istanze utilizzate da implementazioni simultanee che sono in corso per account	us-east-1: 3.000 Tutte le altre regioni supportate: 1.000	Sì	Numero massimo di istanze che possono essere utilizzate da distribuzioni simultane e che sono in corso e che sono associate a un account

Nome	Predefinita	Adatta e	Description
Numero di listener per un instradamento di traffico durante una implementazione Amazon ECS	Ogni regione supportata: 1	No	Numero massimo di listener per un instradamento di traffico durante una implementazione Amazon ECS
Numero di secondi prima che un evento del ciclo di vita di implementazione abbia esito negativo se non viene completato	Ogni regione supportata: 3.600 secondi	No	Numero massimo di secondi prima che un evento del ciclo di vita di una distribuzione abbia esito negativo se non viene completato
Dimensione del nome del gruppo di implementazione	Ogni regione supportata: 100	No	Numero massimo di caratteri per il nome del gruppo di distribuzione
Dimensione della chiave di tag	Ogni regione supportata: 128	No	Numero massimo di caratteri per la chiave di un tag
Dimensione del valore di tag	Ogni regione supportata: 256	No	Numero massimo di caratteri per il valore di un tag
Tag in un gruppo di implementazione	Ogni regione supportata: 10	No	Numero massimo di tag in un gruppo di distribuzione
Traffico che può essere spostato in un unico incremento durante un'implementazione AWS Lambda	Ogni regione supportata: 99	No	Percentuale massima di traffico che può essere spostata in un unico incremento durante un'implementazione AWS Lambda

Risoluzione dei problemi CodeDeploy

Utilizza gli argomenti di questa sezione per risolvere i problemi e gli errori che potresti riscontrare durante l'utilizzo. CodeDeploy

> **Note**
>
> Puoi identificare le cause di molti guasti di distribuzione rivedendo i file di log creati durante il processo di distribuzione. Per semplicità, consigliamo di utilizzare Amazon CloudWatch Logs per monitorare centralmente i file di registro anziché visualizzarli istanza per istanza. Per informazioni, consulta Monitoring Deployments with Amazon CloudWatch Tools.

Argomenti

- Risoluzione di problemi generali
- Risolvi i problemi EC2/On-Premises di distribuzione
- Risolvi i problemi di distribuzione di Amazon ECS
- Risoluzione dei problemi AWS Problemi di implementazione Lambda
- Risoluzione dei problemi relativi ai gruppi di distribuzione
- Risoluzione dei problemi relativi alle istanze
- Risolvi i problemi relativi GitHub ai token
- Risolvi i problemi relativi ad Amazon EC2 Auto Scaling
- Codici di errore per AWS CodeDeploy

Risoluzione di problemi generali

Argomenti

- Lista di controllo per la risoluzione di problemi generali
- CodeDeploy le risorse di distribuzione sono supportate solo in alcune AWS Regioni
- Le procedure descritte in questa guida non corrispondono a quelle della CodeDeploy console
- I ruoli IAM richiesti non sono disponibili
- L'utilizzo di alcuni editor di testo per creare AppSpec file e script di shell può causare il fallimento delle distribuzioni

- L'utilizzo di Finder in macOS per includere in un file compresso la revisione di un'applicazione può determinare l'esito negativo delle distribuzioni

Lista di controllo per la risoluzione di problemi generali

È possibile utilizzare la seguente lista di controllo per la risoluzione dei problemi di distribuzione non riuscita.

1. Consulta Visualizza i dettagli CodeDeploy della distribuzione e View Instance Details per determinare per quale motivo la distribuzione non è riuscita. Se non riesci a determinare la causa, esamina le voci di questo elenco di controllo.
2. Controlla se hai configurato correttamente le istanze:
   - L'istanza è stata lanciata con una coppia di key pair EC2 specificata? Per ulteriori informazioni, consulta EC2 Key Pairs nella Amazon EC2 User Guide.
   - Il profilo di istanza IAM corretto è collegato all'istanza? Per ulteriori informazioni, consultare Configura un'istanza Amazon EC2 con cui lavorare CodeDeploy e Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2.
   - All'istanza sono stati applicati tag? Per ulteriori informazioni, consulta Lavorare con i tag nella console nella Guida per l'utente di Amazon EC2.
   - L' CodeDeploy agente è installato, aggiornato e in esecuzione sull'istanza? Per ulteriori informazioni, consulta Gestione delle operazioni degli CodeDeploy agenti. Per verificare quale versione dell'agente è installata, consultaDetermina la versione dell'agente CodeDeploy.
3. Controlla l'applicazione e le impostazioni del gruppo di distribuzione:
   - Per verificare le impostazioni dell'applicazione, consulta Visualizza i dettagli dell'applicazione con CodeDeploy.
   - Per verificare le impostazioni del gruppo di distribuzione, consulta Visualizza i dettagli del gruppo di distribuzione con CodeDeploy.
4. Conferma che la revisione dell'applicazione sia configurata correttamente:
   - Controllate il formato del AppSpec file. Per ulteriori informazioni, consultare Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy e CodeDeploy AppSpec riferimento al file.
   - Controlla il tuo bucket o GitHub repository Amazon S3 per verificare che la revisione dell'applicazione sia nella posizione prevista.

- Controlla i dettagli della revisione dell' CodeDeploy applicazione per assicurarti che sia registrata correttamente. Per informazioni, consulta Visualizza i dettagli della revisione dell'applicazione con CodeDeploy.
- Se esegui la distribuzione da Amazon S3, controlla il tuo bucket Amazon S3 per CodeDeploy verificare che siano state concesse le autorizzazioni per scaricare la revisione dell'applicazione. Per informazioni sulle policy del bucket, consulta la sezione Prerequisiti di distribuzione.
- Se stai eseguendo la distribuzione da GitHub, controlla il tuo GitHub repository per verificare che siano state concesse le autorizzazioni per scaricare la revisione CodeDeploy dell'applicazione. Per ulteriori informazioni, consultare Crea una distribuzione con CodeDeploy e GitHub autenticazione con applicazioni in CodeDeploy.

5. Controlla se il ruolo del servizio è configurato correttamente. Per informazioni, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.
6. Conferma di aver seguito le fasi in Guida introduttiva con CodeDeploy per:
   - Ha fornito a un utente le autorizzazioni appropriate.
   - Installare o aggiornare e configurare il AWS CLI.
   - Crea un profilo di istanza IAM e un ruolo di servizio.

   Per ulteriori informazioni, consulta Gestione delle identità e degli accessi per AWS CodeDeploy.
7. Conferma di utilizzare la AWS CLI versione 1.6.1 o successiva. Per controllare la versione che hai installato, chiama aws --version.

Se ancora non riesci a risolvere i problemi di distribuzione non riuscita, esaminare gli altri problemi in questo argomento.

CodeDeploy le risorse di distribuzione sono supportate solo in alcune AWS Regioni

Se non vedi o non riesci ad accedere ad applicazioni, gruppi di distribuzione, istanze o altre risorse di distribuzione dalla console AWS CLI o dalla CodeDeploy console, assicurati di fare riferimento a una delle AWS regioni elencate in Regione ed endpoint in. Riferimenti generali di AWS

Le istanze EC2 e i gruppi Amazon EC2 Auto Scaling utilizzati CodeDeploy nelle distribuzioni devono essere avviati e creati in una di queste regioni. AWS

Se utilizzi il, esegui il comando AWS CLI da. aws configure AWS CLI Quindi puoi visualizzare e impostare la tua AWS regione predefinita.

Se utilizzi la CodeDeploy console, nella barra di navigazione, dal selettore Regione, scegli una delle AWS regioni supportate.

> **Important**
>
> Per utilizzare i servizi nella regione Cina (Pechino) o nella regione Cina (Ningxia), è necessario disporre di un account e delle credenziali per tali aree. Gli account e le credenziali di altre AWS regioni non funzionano per le regioni di Pechino e Ningxia e viceversa.
> Le informazioni su alcune risorse per le regioni della Cina, come i nomi dei bucket del CodeDeploy Resource Kit e le procedure di installazione degli CodeDeploy agenti, non sono incluse in questa edizione della Guida per l'CodeDeploy utente.
> Per ulteriori informazioni:
>
> - CodeDeployin Guida introduttiva AWS nella regione Cina (Pechino)
> - CodeDeploy Guida per l'utente per le regioni della Cina (versione inglese | versione cinese)

Le procedure descritte in questa guida non corrispondono a quelle della CodeDeploy console

Le procedure in questa guida sono scritte in base al nuovo progetto di console. Tuttavia molti dei concetti e delle procedure di base in questa guida si applicano anche alla versione precedente della console. Per accedere alla guida nella nuova console, scegli l'icona delle informazioni.

I ruoli IAM richiesti non sono disponibili

Se ti affidi a un profilo di istanza IAM o a un ruolo di servizio creato come parte di uno AWS CloudFormation stack, se elimini lo stack, vengono eliminati anche tutti i ruoli IAM. Questo potrebbe essere il motivo per cui il ruolo IAM non viene più visualizzato nella console IAM e CodeDeploy non funziona più come previsto. Per risolvere questo problema, devi ricreare manualmente il ruolo IAM eliminato.

L'utilizzo di alcuni editor di testo per creare AppSpec file e script di shell può causare il fallimento delle distribuzioni

Alcuni editor di testo introducono caratteri non conformi e non stampabili nei file. Se utilizzi editor di testo per creare o modificare AppSpec file o file di script di shell da eseguire su istanze Amazon Linux, Ubuntu Server o RHEL, qualsiasi distribuzione basata su questi file potrebbe fallire. Quando

CodeDeploy utilizza questi file durante una distribuzione, la presenza di questi caratteri può portare a errori di convalida dei file e di esecuzione degli script difficili da risolvere AppSpec .

Nella CodeDeploy console, nella pagina dei dettagli dell'evento per la distribuzione, scegli Visualizza registri. (Oppure puoi usare il AWS CLI per chiamare il comando get-deployment-instance.) Cerca gli errori, ad esempio `invalid character`, `command not found` o `file not found`.

Per risolvere questo problema, è consigliabile attenersi alle seguenti indicazioni:

- Non utilizzate editor di testo che introducono caratteri non stampabili come i caratteri di ritorno (`^M`caratteri) nei file e nei file di script di shell. AppSpec
- Utilizzate editor di testo che visualizzano caratteri non stampabili, come i riage return nei file e i AppSpec file di shell script, in modo da poter trovare e rimuovere tutti quelli che potrebbero essere introdotti. Per esempi di questi tipi di editor di testo, esegui una ricerca su Internet dei termini "editor di testo che mostrano ritorni a capo".
- Usa editor di testo in esecuzione su istanze Amazon Linux, Ubuntu Server o RHEL per creare file di script di shell da eseguire su istanze Amazon Linux, Ubuntu Server o RHEL. Per esempi di questi tipi di editor di testo, esegui una ricerca su Internet dei termini "editor script shell Linux".
- Se devi usare un editor di testo in Windows o macOS per creare file di script di shell da eseguire su istanze Amazon Linux, Ubuntu Server o RHEL, usa un programma o un'utilità che converta il testo in formato Windows o macOS in formato Unix. Per esempi di questi programmi e utilità, esegui una ricerca su Internet dei termini "da DOS a UNIX" o "da Mac a UNIX". Assicurati di testare i file di script shell convertiti sui sistemi operativi di destinazione.

L'utilizzo di Finder in macOS per includere in un file compresso la revisione di un'applicazione può determinare l'esito negativo delle distribuzioni

Le distribuzioni potrebbero non riuscire se si utilizza l'applicazione dell'interfaccia grafica utente (GUI) Finder su un Mac per raggruppare (comprimere) un AppSpec file e i file e gli script correlati in un file di archivio di revisione dell'applicazione (.zip). Ciò si verifica perché Finder crea una cartella `__MACOSX` intermedia nel file.zip e vi colloca al suo interno i file dei componenti. CodeDeploy non è in grado di individuare i file dei componenti, per cui la distribuzione ha esito negativo.

Per risolvere questo problema, ti consigliamo di utilizzare il comando AWS CLI to call the push, che comprime i file dei componenti nella struttura prevista. In alternativa, puoi utilizzare Terminal anziché la GUI per comprimere i file dei componenti in formato zip. Terminal non crea una cartella `__MACOSX` intermedia.

Risolvi i problemi EC2/On-Premises di distribuzione

Argomenti

- CodeDeploy errore relativo alle credenziali mancanti del plugin CommandPoller
- La distribuzione non riesce con il messaggio «Convalida del messaggio firmato PKCS7 fallita»
- La distribuzione o la ridistribuzione degli stessi file alle stesse posizioni delle istanze avrà esito negativo e mostrerà l'errore "La distribuzione non è riuscita a causa di un file specificato già esistente in questa posizione"
- I percorsi di file lunghi causano errori del tipo «Nessun file o directory di questo tipo»
- Long-running i processi possono causare il fallimento delle distribuzioni
- Risoluzione dei problemi relativi a un evento AllowTraffic del ciclo di vita non riuscito senza che sia stato riportato alcun errore nei log di distribuzione
- Risoluzione dei problemi relativi a un evento non riuscito o ApplicationStop AfterBlockTraffic relativo al BeforeBlockTraffic ciclo di vita dell'implementazione
- Risoluzione dei problemi relativi a un evento del ciclo di vita di DownloadBundle distribuzione non riuscito con UnknownError: not open-for reading
- Risoluzione dei problemi relativi a tutti gli eventi del ciclo di vita ignorati
- Per impostazione predefinita, PowerShell gli script di Windows non riescono a utilizzare la versione a 64 bit di Windows PowerShell

> **Note**
>
> Le cause di molti guasti di distribuzione possono essere identificate rivedendo i file di log creati durante il processo di distribuzione. Per semplicità, consigliamo di utilizzare Amazon CloudWatch Logs per monitorare centralmente i file di registro anziché visualizzarli istanza per istanza. Per informazioni, consulta View CodeDeploy Logs in CloudWatch Logs Console.

> **Tip**
>
> Per un runbook che automatizza molte attività di risoluzione dei problemi relative alle EC2/On-Premises distribuzioni, vedere il riferimento al runbook di riferimento di AWSSupport-TroubleshootCodeDeploy AWSSystems Manager Automation.

CodeDeploy errore relativo alle credenziali mancanti del plugin CommandPoller

Se si verifica un errore simile a `InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller: Missing credentials - please check if this instance was started with an IAM instance profile`, il motivo potrebbe essere uno dei seguenti:

- L'istanza su cui stai eseguendo la distribuzione non ha un profilo di istanza IAM associato.
- Il tuo profilo di istanza IAM non dispone delle autorizzazioni corrette configurate.

Un profilo di istanza IAM concede all' CodeDeploy agente l'autorizzazione a comunicare CodeDeploy e scaricare la revisione da Amazon S3. Per le istanze EC2, consulta Gestione delle identità e degli accessi per AWS CodeDeploy. Per le istanze locali, consulta Working with On-Premises Instances.

La distribuzione non riesce con il messaggio «Convalida del messaggio firmato PKCS7 fallita»

Questo messaggio di errore indica che l'istanza sta eseguendo una versione dell' CodeDeploy agente che supporta solo l' SHA-1 algoritmo hash. Il supporto per l'algoritmo SHA-2 hash è stato introdotto nella versione 1.0.1.854 dell' CodeDeploy agente, rilasciata a novembre 2015. A partire dal 17 ottobre 2016, le distribuzioni non riescono se viene installata una versione dell'agente precedente alla CodeDeploy 1.0.1.854. Per ulteriori informazioni, vedere per passare AWS all'algoritmo hash SHA256 per i certificati SSL, NOTA: Ritiro degli agenti host precedenti alla versione 1.0.1.85 e. CodeDeploy Aggiorna l' CodeDeploy agente

La distribuzione o la ridistribuzione degli stessi file alle stesse posizioni delle istanze avrà esito negativo e mostrerà l'errore "La distribuzione non è riuscita a causa di un file specificato già esistente in questa posizione"

Quando si CodeDeploy tenta di distribuire un file su un'istanza ma un file con lo stesso nome esiste già nella posizione di destinazione specificata, la distribuzione su quell'istanza potrebbe non riuscire. Potresti ricevere il messaggio di errore «La distribuzione non è riuscita perché un file specificato esiste già in questa posizione:*`location-name`*». Ciò si verifica perché, durante ciascuna distribuzione, CodeDeploy prima elimina tutti i file dalla distribuzione precedente, che sono elencati in un file di log di pulizia. Se nelle cartelle di installazione di destinazione sono presenti file che non

sono elencati in questo file di pulizia, l' CodeDeploy agente per impostazione predefinita lo interpreta come un errore e fallisce la distribuzione.

> **Note**
>
> Nelle istanze di Amazon Linux, RHEL e Ubuntu Server, il file di pulizia si trova in. `/opt/codedeploy-agent/deployment-root/deployment-instructions/` Nelle istanze di Windows Server, la posizione è. `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\`

Il modo più semplice per evitare questo errore è specificare un'opzione diversa dal comportamento predefinito per l'esito negativo della distribuzione. Per ciascuna distribuzione, è possibile scegliere se far sì che la distribuzione abbia esito negativo, sovrascrivere i file non inclusi nel file di pulizia o conservare i file già presenti nell'istanza.

L'opzione di sovrascrittura è utile, ad esempio, se collochi manualmente un file in un'istanza dopo l'ultima distribuzione e poi aggiungi un file con lo stesso nome nella successiva revisione dell'applicazione.

Potresti scegliere l'opzione di conservazione per i file nell'istanza che desideri inserire nella successiva distribuzione senza doverli aggiungere al pacchetto della revisione dell'applicazione. L'opzione retention è utile anche se i file dell'applicazione si trovano già nell'ambiente di produzione e si desidera distribuirli utilizzando CodeDeploy per la prima volta. Per ulteriori informazioni, consultare Crea una distribuzione di EC2/On-Premises Compute Platform (console) e Comportamento di rollback con il contenuto esistente.

Risoluzione dei problemi **`La distribuzione non è riuscita perché in questa posizione esiste già un file specificato`** (errori di distribuzione).

Se scegli di non specificare un'opzione per sovrascrivere o conservare i contenuti che CodeDeploy rileva nelle tue posizioni di distribuzione di destinazione (o se non specifichi alcuna opzione di distribuzione per la gestione dei contenuti esistenti in un comando programmatico), puoi scegliere di eseguire la risoluzione dell'errore.

Le informazioni seguenti si applicano solo se scegli di non conservare o sovrascrivere i contenuti.

Se si tenta di ridistribuire file con gli stessi nomi e posizioni, è più probabile che la ridistribuzione abbia successo se si specificano il nome dell'applicazione e il gruppo di distribuzione con lo stesso ID del gruppo di distribuzione sottostante utilizzato in precedenza. CodeDeploy utilizza l'ID del gruppo di distribuzione sottostante per identificare i file da rimuovere prima di una ridistribuzione.

La distribuzione di nuovi file o la ridistribuzione degli stessi file nelle stesse posizioni sulle istanze può avere esito negativo per questi motivi:

- Hai specificato un nome dell'applicazione diverso per una ridistribuzione della stessa revisione alle stesse istanze. La ridistribuzione ha esito negativo perché, anche se il nome del gruppo di distribuzione è lo stesso, l'uso di un nome dell'applicazione diverso comporta l'utilizzo di un ID gruppo di distribuzione sottostante differente.
- Hai eliminato e ricreato un gruppo di distribuzione per un'applicazione e poi hai tentato di ridistribuire la stessa revisione al gruppo di distribuzione. La ridistribuzione non riesce perché, anche se il nome del gruppo di distribuzione è lo stesso, CodeDeploy fa riferimento a un ID del gruppo di distribuzione sottostante diverso.
- Hai eliminato un'applicazione e un gruppo di distribuzione in CodeDeploy, quindi hai creato una nuova applicazione e un gruppo di distribuzione con gli stessi nomi di quelli eliminati. Dopodiché hai tentato di ridistribuire al nuovo gruppo di distribuzione una revisione che era stata distribuita al gruppo di distribuzione precedente con lo stesso nome. La ridistribuzione non riesce perché, anche se i nomi dell'applicazione e del gruppo di distribuzione sono gli stessi, fa CodeDeploy comunque riferimento all'ID del gruppo di distribuzione eliminato.
- Hai distribuito una revisione a un gruppo di distribuzione e poi hai distribuito la stessa revisione a un altro gruppo di distribuzione per le stesse istanze. La seconda distribuzione ha esito negativo perché CodeDeploy fa riferimento a un ID gruppo di distribuzione sottostante differente.
- Hai distribuito una revisione a un gruppo di distribuzione e poi hai distribuito un'altra revisione a un altro gruppo di distribuzione per le stesse istanze. Vi è almeno un file con lo stesso nome e nella stessa posizione che il secondo gruppo di distribuzione cerca di distribuire. La seconda distribuzione non riesce perché CodeDeploy non rimuove il file esistente prima dell'inizio della seconda distribuzione. Entrambe le distribuzioni fanno riferimento a ID di gruppi di distribuzione diversi.
- Hai distribuito una revisione in CodeDeploy, ma esiste almeno un file con lo stesso nome e nella stessa posizione. La distribuzione non riesce perché, per impostazione predefinita, CodeDeploy non rimuove il file esistente prima dell'inizio della distribuzione.

Per risolvere queste situazioni, procedi in uno dei seguenti modi:

- Rimuovi i file dalle posizioni e dalle istanze in cui erano stati distribuiti in precedenza e prova a eseguire nuovamente la distribuzione.
- Nel AppSpec file della revisione, negli eventi del ciclo di vita della BeforeInstall distribuzione ApplicationStop o della distribuzione, specifica uno script personalizzato per eliminare i file in tutte le posizioni che corrispondono ai file che la revisione sta per installare.
- Distribuisci o ridistribuisci i file a posizioni o istanze che non erano parte di distribuzioni precedenti.
- Prima di eliminare un'applicazione o un gruppo di distribuzione, distribuite una revisione che contenga un AppSpec file che specifichi l'assenza di file da copiare nelle istanze. Per la distribuzione, specifica il nome dell'applicazione e il nome del gruppo di distribuzione che utilizzano gli stessi ID dell'applicazione e del gruppo di distribuzione sottostanti che stai per eliminare. (È possibile utilizzare il comando get-deployment-group per recuperare l'ID del gruppo di distribuzione.) CodeDeployutilizza l'ID e il file del gruppo di distribuzione sottostanti per rimuovere tutti i AppSpec file installati nella precedente distribuzione riuscita.

I percorsi di file lunghi causano errori del tipo «Nessun file o directory di questo tipo»

Per le distribuzioni su istanze di Windows, se nella sezione file del file appspec.yml è presente un percorso di oltre 260 caratteri, è possibile che le distribuzioni non riescano con un errore simile al seguente:

`No such file or directory @ dir_s_mkdir - C:\`*`your-long-file-path`*

Questo errore si verifica perché Windows per impostazione predefinita non consente percorsi di file superiori a 260 caratteri, come dettagliato nella documentazione Microsoft.

Per le versioni CodeDeploy dell'agente 1.4.0 o successive, è possibile abilitare i percorsi di file lunghi in due modi, a seconda del processo di installazione dell'agente:

Se l' CodeDeploy agente non è ancora stato installato:

1. Sul computer in cui intendi installare l' CodeDeploy agente, abilita la chiave di registro di `LongPathsEnabled` Windows utilizzando questo comando:

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"
          -Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Installa l' CodeDeploy agente. Per ulteriori informazioni, consulta Installa l' CodeDeploy agente.

Se l' CodeDeploy agente è già stato installato:

1. Sul computer dell' CodeDeploy agente, abilita la chiave di registro di `LongPathsEnabled` Windows utilizzando questo comando:

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"
-Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Riavviare l' CodeDeploy agente per rendere effettiva la modifica della chiave di registro. Per riavviare l'agente, utilizzate questo comando:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Long-running i processi possono causare il fallimento delle distribuzioni

Per le distribuzioni su istanze Amazon Linux, Ubuntu Server e RHEL, se disponi di uno script di distribuzione che avvia un processo di lunga durata, CodeDeploy potresti attendere a lungo l'evento del ciclo di vita della distribuzione e poi fallire la distribuzione. Ciò accade perché, se il processo viene eseguito per un periodo di tempo maggiore rispetto a quello previsto per i processi in primo piano e in background di quell'evento, CodeDeploy si interrompe e la distribuzione non riesce, anche se il processo viene comunque eseguito come previsto.

Ad esempio, una revisione dell'applicazione contiene due file nella propria radice, `after-install.sh` e `sleep.sh`. Il file contiene le seguenti istruzioni: AppSpec

```
version: 0.0
os: linux
files:
  - source: ./sleep.sh
    destination: /tmp
hooks:
  AfterInstall:
    - location: after-install.sh
      timeout: 60
```

Il `after-install.sh` file viene eseguito durante l'evento del ciclo di vita AfterInstall dell'applicazione. Di seguito sono descritti i contenuti:

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

```
/tmp/sleep.sh
```

Il file `sleep.sh` include il contenuto seguente, che sospende l'esecuzione del programma per tre minuti (180 secondi), simulando un processo di lunga durata:

```
#!/bin/bash
sleep 180
```

In caso di `after-install.sh` chiamata`sleep.sh`, viene `sleep.sh` avviato ed eseguito per tre minuti (180 secondi), ovvero due minuti (120 secondi) dopo l'orario CodeDeploy previsto `sleep.sh` (e, per relazione,`after-install.sh`) l'interruzione dell'esecuzione. Dopo il timeout di un minuto (60 secondi), CodeDeploy interrompe e fallisce la distribuzione all'evento del ciclo di vita AfterInstall dell'applicazione, anche se `sleep.sh` continua a funzionare come previsto. Viene visualizzato l'errore seguente:

`Script at specified location: after-install.sh failed to complete in 60 seconds`.

Non è sufficiente aggiungere una e commerciale (`&`) in `after-install.sh` per eseguire `sleep.sh` in background.

```
#!/bin/bash
# Do not do this.
/tmp/sleep.sh &
```

In questo modo è possibile lasciare la distribuzione in sospeso fino al periodo di timeout predefinito di un'ora per l'evento del ciclo di vita dell'implementazione, dopodiché CodeDeploy interrompe e fallisce la distribuzione in occasione dell'evento del ciclo di vita dell'applicazione come in precedenza. AfterInstall

In`after-install.sh`, chiama `sleep.sh` come segue, il che consente CodeDeploy di continuare dopo l'avvio del processo:

```
#!/bin/bash
/tmp/sleep.sh > /dev/null 2> /dev/null < /dev/null &
```

Nella chiamata precedente `sleep.sh` è il nome del processo la cui esecuzione deve essere avviata in background, con il reindirizzamento di stdout, stderr e stdin a `/dev/null`.

Risoluzione dei problemi relativi a un evento AllowTraffic del ciclo di vita non riuscito senza che sia stato riportato alcun errore nei log di distribuzione

In alcuni casi, una blue/green distribuzione non riesce durante l'evento del AllowTraffic ciclo di vita, ma i registri di distribuzione non indicano la causa dell'errore.

Questo errore è in genere dovuto a controlli di integrità configurati in modo errato in Elastic Load Balancing for the Classic Load Balancer, Application Load Balancer o Network Load Balancer utilizzati per gestire il traffico per il gruppo di distribuzione.

Per risolvere il problema, esamina e correggi eventuali errori nella configurazione del controllo dello stato per il sistema di bilanciamento del carico.

Per i Classic Load Balancer, consulta Configure Health Checks nella User Guide for Classic Load Balancers e ConfigureHealthCheck nella versione di riferimento dell'API Elastic Load Balancing 2012-06-01.

Per gli Application Load Balancer, consulta Health Checks for Your Target Groups nella User Guide for Application Load Balancers.

Per i Network Load Balancer, consulta Health Checks for Your Target Groups nella Network Load Balancer User Guide.

Risoluzione dei problemi relativi a un evento non riuscito o ApplicationStop AfterBlockTraffic relativo al BeforeBlockTraffic ciclo di vita dell'implementazione

Durante una distribuzione, l' CodeDeploy agente esegue gli script specificati per ApplicationStop e BeforeBlockTraffic contenuti AfterBlockTraffic nel AppSpec file della precedente distribuzione riuscita. (Tutti gli altri script vengono eseguiti dal AppSpec file nella distribuzione corrente.) Se uno di questi script contiene un errore e non viene eseguito correttamente, è possibile che la distribuzione non riesca.

Le possibili cause di questa situazione includono:

- L' CodeDeploy agente trova il *deployment-group-id*_last_successful_install file nella posizione corretta, ma la posizione elencata nel *deployment-group-id*_last_successful_install file non esiste.

Nelle istanze Amazon Linux, Ubuntu Server e RHEL, questo file deve esistere in. `/opt/codedeploy-agent/deployment-root/deployment-instructions`

Nelle istanze di Windows Server, questo file deve essere archiviato nella cartella. `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions`

- Nella posizione elencata nel *`deployment-group-id`*`_last_successful_install` file, il AppSpec file non è valido o gli script non vengono eseguiti correttamente.
- Lo script contiene un errore che non può essere corretto, pertanto non verrà mai eseguito correttamente.

Usa la CodeDeploy console per indagare sul motivo per cui una distribuzione potrebbe non essere riuscita durante uno di questi eventi. Nella pagina dei dettagli per la distribuzione scegliere View events (Visualizza eventi). Nella pagina dei dettagli dell'istanza, nella AfterBlockTrafficriga ApplicationStopBeforeBlockTraffic, o, scegli Visualizza registri. Oppure usa AWS CLI per chiamare il comando get-deployment-instance.

Se la causa dell'errore è uno script dell'ultima distribuzione riuscita che non viene mai eseguito correttamente, crea una distribuzione e specifica che gli errori ApplicationStop BeforeBlockTraffic, e devono essere ignorati. AfterBlockTraffic Ci sono due modi per effettuare questa operazione:

- Usa la CodeDeploy console per creare una distribuzione. Nella pagina Crea distribuzione, in Errore dell'evento ApplicationStop del ciclo di vita, scegli Non interrompere la distribuzione su un'istanza se questo evento del ciclo di vita sull'istanza fallisce.
- Usa AWS CLI per chiamare il create-deployment comando e includere l'opzione. `--ignore-application-stop-failures`

Quando si distribuisce nuovamente la revisione dell'applicazione, la distribuzione continua anche se uno di questi tre eventi del ciclo di vita non va a buon fine. Se la nuova revisione include script fissi per quegli eventi del ciclo di vita, è possibile che le distribuzioni future riescano senza applicare questa correzione.

Risoluzione dei problemi relativi a un evento del ciclo di vita di DownloadBundle distribuzione non riuscito con UnknownError: not open-for reading

Se stai tentando di distribuire una revisione dell'applicazione da Amazon S3 e la distribuzione non riesce durante l'evento del ciclo di vita DownloadBundle della distribuzione con l'errore: `UnknownError: not opened for reading`

- Si è verificato un errore interno del servizio Amazon S3. Distribuisci nuovamente la revisione dell'applicazione.
- Il profilo dell'istanza IAM sulla tua istanza EC2 non dispone delle autorizzazioni per accedere alla revisione dell'applicazione in Amazon S3. Per informazioni sulle policy dei bucket di Amazon S3, consulta e. Invia una revisione CodeDeploy ad Amazon S3 EC2/On-Premises (solo distribuzioni) Prerequisiti di distribuzione
- Le istanze in cui esegui la distribuzione sono associate a una AWS regione (ad esempio, Stati Uniti occidentali (Oregon)), ma il bucket Amazon S3 che contiene la revisione dell'applicazione è associato a un'altra AWS regione (ad esempio, Stati Uniti orientali (Virginia settentrionale)). Assicurati che la revisione dell'applicazione si trovi in un bucket Amazon S3 associato alla AWS stessa regione delle istanze.

Nella pagina dei dettagli degli eventi per la distribuzione, nella riga Download bundle (Scarica bundle), scegliere View logs (Visualizza log). Oppure usa per chiamare il comando AWS CLI get-deployment-instance. Se si verifica, l'errore deve essere indicato nell'output con il codice di errore `UnknownError` e il messaggio di errore `not opened for reading`.

Per stabilire il motivo di questo errore:

1. Abilitare la registrazione in rete su almeno una delle istanze, quindi distribuire nuovamente la revisione dell'applicazione.
2. Esaminare il file di registrazione in rete per trovare l'errore. I messaggi di errore più comuni per questo problema includono la frase "access denied" (accesso negato).
3. Dopo avere esaminato i file di log, è consigliabile disabilitare la registrazione in rete per ridurre la dimensione del file di log e la quantità di informazioni sensibili che in futuro potrebbero comparire nell'output in testo normale nell'istanza.

Per informazioni su come trovare il file di registrazione dei cavi e abilitare e disabilitare la registrazione via cavo, vedere nella guida di riferimento alla configurazione dell'agente. `:log_aws_wire:` CodeDeploy

Risoluzione dei problemi relativi a tutti gli eventi del ciclo di vita ignorati

Se tutti gli eventi del ciclo di vita di una distribuzione EC2 o locale vengono ignorati, potresti ricevere un errore simile a. `The overall deployment failed because too many individual instances failed deployment, too few healthy instances are available for deployment, or some instances in your deployment group are experiencing problems. (Error code: HEALTH_CONSTRAINTS)` Di seguito sono elencate alcune possibili cause e soluzioni:

- L' CodeDeploy agente potrebbe non essere installato o in esecuzione sull'istanza. Per determinare se l' CodeDeploy agente è in esecuzione:
  - Per il server Amazon Linux RHEL o Ubuntu, esegui:

    ```
    systemctl status codedeploy-agent
    ```

  - Per Windows, esegui:

    ```
    powershell.exe -Command Get-Service -Name CodeDeployagent
    ```

  Se l' CodeDeploy agente non è installato o non è in esecuzione, vedereVerifica che l' CodeDeploy agente sia in esecuzione.

  L'istanza potrebbe non essere in grado di raggiungere l'endpoint pubblico CodeDeploy o Amazon S3 utilizzando la porta 443. Provare con una delle seguenti operazioni:

  - Assegna un indirizzo IP pubblico all'istanza e utilizza la tabella di routing per consentire l'accesso a Internet. Verifica che il gruppo di sicurezza associato all'istanza consenta l'accesso in uscita sulla porta 443 (HTTPS). Per ulteriori informazioni, consulta Protocollo di comunicazione e porta per l'agente CodeDeploy.
  - Se un'istanza viene assegnata in una sottorete privata, utilizza un gateway NAT anziché un gateway Internet nella tabella di routing. Per ulteriori informazioni, consulta Gateway NAT.
- Il ruolo di servizio per CodeDeploy potrebbe non avere le autorizzazioni richieste. Per configurare un ruolo del servizio CodeDeploy, consulta Fase 2: Creare un ruolo di servizio per CodeDeploy.

- Se utilizzate un proxy HTTP, assicuratevi che sia specificato nell'`:proxy_uri:`impostazione del file di configurazione dell' CodeDeploy agente. Per ulteriori informazioni, consulta CodeDeploy riferimento alla configurazione dell'agente.
- La data e l'ora dell'istanza di distribuzione potrebbero non corrispondere alla data e all'ora della richiesta di distribuzione. Cerca un errore simile a quello presente `Cannot reach InstanceService: Aws::CodeDeployCommand::Errors::InvalidSignatureException - Signature expired` nel file di registro CodeDeploy dell'agente. Se l'errore è presente, segui i passaggi descritti in Risoluzione dei problemi relativi alla distribuzione «InvalidSignatureException — Firma scaduta: [ora] è precedente a [ora]». Per ulteriori informazioni, consulta Visualizza i dati di registro per le CodeDeploy EC2/On-Premises distribuzioni.
- L' CodeDeploy agente potrebbe smettere di funzionare perché un'istanza sta esaurendo la memoria o lo spazio su disco rigido. Prova a ridurre il numero di distribuzioni archiviate sulla tua istanza aggiornando l'`max_revisions`impostazione nella configurazione dell'agente. CodeDeploy Se esegui questa operazione per un'istanza EC2 e il problema persiste, prendi in considerazione l'utilizzo di un'istanza più grande. Ad esempio, se il tipo di istanza è `t2.small`, prova a usare un'stanza `t2.medium`. Per ulteriori informazioni, consulta File installati dall'agente CodeDeployCodeDeploy riferimento alla configurazione dell'agente, e Tipi di istanze.
- L'istanza su cui stai eseguendo la distribuzione potrebbe non avere un profilo di istanza IAM collegato o potrebbe avere un profilo di istanza IAM collegato che non dispone delle autorizzazioni richieste.
  - Se un profilo di istanza IAM non è collegato alla tua istanza, creane uno con le autorizzazioni richieste e poi collegalo.
  - Se un profilo di istanza IAM è già collegato alla tua istanza, assicurati che disponga delle autorizzazioni richieste.

  Dopo aver verificato che il profilo di istanza collegato è configurato con le autorizzazioni richieste, riavvia l'istanza. Per ulteriori informazioni, consulta Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2 IAM Roles for Amazon EC2 nella Amazon EC2 User Guide.

Per impostazione predefinita, PowerShell gli script di Windows non riescono a utilizzare la versione a 64 bit di Windows PowerShell

Se uno PowerShell script di Windows eseguito come parte di una distribuzione si basa sulla funzionalità a 64 bit (ad esempio, perché consuma più memoria di quella consentita da

un'applicazione a 32 bit o chiama librerie offerte solo in una versione a 64 bit), lo script potrebbe bloccarsi o non essere eseguito come previsto. Questo perché, per impostazione predefinita, CodeDeploy utilizza la versione a 32 bit di Windows PowerShell per eseguire PowerShell script Windows che fanno parte della revisione di un'applicazione.

Aggiungi codice come il seguente all'inizio di qualsiasi script che deve essere eseguito con la versione a 64 bit di Windows: PowerShell

```
# Are you running in 32-bit mode?
#   (\SysWOW64\ = 32-bit mode)

if ($PSHOME -like "*SysWOW64*")
{
  Write-Warning "Restarting this script under 64-bit Windows PowerShell."

  # Restart this script under 64-bit Windows PowerShell.
  #   (\SysNative\ redirects to \System32\ for 64-bit mode)

  & (Join-Path ($PSHOME -replace "SysWOW64", "SysNative") powershell.exe) -File `
    (Join-Path $PSScriptRoot $MyInvocation.MyCommand) @args

  # Exit 32-bit script.

  Exit $LastExitCode
}

# Was restart successful?
Write-Warning "Hello from $PSHOME"
Write-Warning "  (\SysWOW64\ = 32-bit mode, \System32\ = 64-bit mode)"
Write-Warning "Original arguments (if any): $args"

# Your 64-bit script code follows here...
# ...
```

Sebbene le informazioni sul percorso dei file contenute in questo codice possano sembrare controintuitive, Windows a 32 bit PowerShell utilizza un percorso come:

`c:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe`

Windows a 64 bit PowerShell utilizza un percorso come:

`c:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`

Risolvi i problemi di distribuzione di Amazon ECS

Argomenti

- Si verifica un timeout durante l'attesa del set di attività sostitutive
- Si verifica un timeout in attesa del proseguimento di una notifica
- Il ruolo IAM non dispone di autorizzazioni sufficienti
- La distribuzione è scaduta in attesa di una richiamata di stato
- La distribuzione non è riuscita perché una o più delle funzioni di convalida degli eventi del ciclo di vita non sono riuscite
- L'ELB non può essere aggiornato a causa del seguente errore: Il gruppo target del taskset primario deve essere dietro l'ascoltatore
- La mia distribuzione a volte non riesce quando utilizzo Auto Scaling
- Solo ALB supporta il routing graduale del traffico, utilizza invece il routing AllAtOnce del traffico quando crei un gruppo di distribuzione create/update
- Anche se la mia distribuzione è riuscita, il set di attività sostitutivo non supera i controlli di integrità di Elastic Load Balancing e la mia applicazione è inattiva
- Posso collegare più sistemi di bilanciamento del carico a un gruppo di distribuzione?
- Posso eseguire CodeDeploy blue/green implementazioni senza un sistema di bilanciamento del carico?
- Come posso aggiornare il mio servizio Amazon ECS con nuove informazioni durante una distribuzione?

Si verifica un timeout durante l'attesa del set di attività sostitutive

Problema: viene visualizzato il seguente messaggio di errore durante la distribuzione dell'applicazione Amazon ECS utilizzando: CodeDeploy

```
The deployment timed out while waiting for the replacement task set to
become healthy. This time out period is 60 minutes.
```

Possibile causa: questo errore può verificarsi se è presente un errore nel file di definizione delle attività o in altri file relativi alla distribuzione. Ad esempio, se c'è un errore di battitura nel `image` campo del file di definizione delle attività, Amazon ECS tenterà di estrarre l'immagine del contenitore sbagliata e fallirà continuamente, provocando questo errore.

Possibili correzioni e passaggi successivi:

- Correggi gli errori tipografici e i problemi di configurazione nel file di definizione delle attività e in altri file.
- Controlla l'evento relativo al servizio Amazon ECS e scopri perché le attività sostitutive non stanno migliorando. Per ulteriori informazioni sugli eventi Amazon ECS, consulta gli eventi Amazon ECS nella Amazon Elastic Container Service Developer Guide.
- Consulta la sezione sulla risoluzione dei problemi di Amazon ECS nella Amazon Elastic Container Service Developer Guide per eventuali errori relativi ai messaggi presenti nell'evento.

Si verifica un timeout in attesa del proseguimento di una notifica

Problema: viene visualizzato il seguente messaggio di errore durante la distribuzione dell'applicazione Amazon ECS utilizzando: CodeDeploy

`The deployment timed out while waiting for a notification to continue. This time out period is` *`n`* `minutes.`

Possibile causa: questo errore può verificarsi se hai specificato un tempo di attesa nel campo Specifica quando reindirizzare il traffico al momento della creazione del gruppo di distribuzione, ma la distribuzione non è stata completata prima della scadenza del tempo di attesa.

Possibili correzioni e passaggi successivi:

- Nel tuo gruppo di distribuzione, imposta Specifica quando reindirizzare il traffico su un periodo di tempo maggiore e ridistribuirlo. Per ulteriori informazioni, consulta Crea un gruppo di distribuzione per una distribuzione Amazon ECS (console).
- Nel tuo gruppo di distribuzione, modifica Specifica quando reindirizzare il traffico su Reindirizza immediatamente il traffico e ridistribuiscilo. Per ulteriori informazioni, consulta Crea un gruppo di distribuzione per una distribuzione Amazon ECS (console).
- Ridistribuisci ed esegui il comando aws deploy continue-deployment con l'opzione impostata su. AWS CLI `--deployment-wait-type READY_WAIT` Assicurati di eseguire questo comando prima della scadenza del tempo specificato in Specifica quando il traffico deve essere reindirizzato.

Il ruolo IAM non dispone di autorizzazioni sufficienti

Problema: viene visualizzato il seguente messaggio di errore durante la distribuzione dell'applicazione Amazon ECS utilizzando: CodeDeploy

`The IAM role role-arn does not give you permission to perform operations in the following AWS service: AWSLambda.`

Possibile causa: questo errore può verificarsi se hai specificato una funzione Lambda nella `Hooks` sezione del AppSpec file, ma non hai dato l' CodeDeploy autorizzazione al servizio Lambda.

Possibile soluzione: aggiungere l'`lambda:InvokeFunction`autorizzazione al ruolo del CodeDeploy servizio. Per aggiungere questa autorizzazione, aggiungi una delle seguenti politiche AWS gestite al ruolo: **`AWSCodeDeployRoleForECS`** o**`AWSCodeDeployRoleForECSLimited`**. Per informazioni su queste politiche e su come aggiungerle al ruolo CodeDeploy di servizio, vedereFase 2: Creare un ruolo di servizio per CodeDeploy.

La distribuzione è scaduta in attesa di una richiamata di stato

Problema: viene visualizzato il seguente messaggio di errore durante la distribuzione dell'applicazione Amazon ECS utilizzando: 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.`

Possibile causa: questo errore può verificarsi se hai specificato una funzione Lambda nella `Hooks` sezione del AppSpec file, ma la funzione Lambda non è riuscita a chiamare l'`PutLifecycleEventHookExecutionStatus`API necessaria a cui restituire uno `Succeeded` stato or. `Failed` CodeDeploy

Possibili correzioni e passaggi successivi:

- Aggiungi l'`codedeploy:putlifecycleEventHookExecutionStatus`autorizzazione al ruolo di esecuzione Lambda utilizzato dalla funzione Lambda specificata nel file. AppSpec Questa autorizzazione concede alla funzione Lambda la possibilità di restituire uno stato pari o a. `Succeeded` `Failed` CodeDeploy Per ulteriori informazioni sul ruolo di esecuzione Lambda, consulta il ruolo di esecuzione Lambda nella Guida per l'utente. AWS Lambda
- Controlla il codice della funzione Lambda e i log di esecuzione per assicurarti che la funzione Lambda stia chiamando l'`PutLifecycleEventHookExecutionStatus`API

per indicare se CodeDeploy il test di convalida del ciclo CodeDeploy di vita o. `Succeeded` `Failed` Per informazioni sull'`putlifecycleEventHookExecutionStatus`API, consulta PutLifecycleEventHookExecutionStatus l'API Reference. AWS CodeDeploy Per informazioni sui log di esecuzione di Lambda, consulta Accessing Amazon CloudWatch logs for AWS Lambda.

La distribuzione non è riuscita perché una o più delle funzioni di convalida degli eventi del ciclo di vita non sono riuscite

Problema: viene visualizzato il seguente messaggio di errore durante la distribuzione dell'applicazione Amazon ECS utilizzando: CodeDeploy

```
The deployment failed because one or more of the lifecycle event validation functions failed.
```

Possibile causa: questo errore può verificarsi se hai specificato una funzione Lambda nella `Hooks` sezione del AppSpec file, ma la funzione Lambda è tornata `Failed` al CodeDeploy momento della chiamata. `PutLifecycleEventHookExecutionStatus` Questo errore indica CodeDeploy che il test di convalida del ciclo di vita non è riuscito.

Possibile passaggio successivo: controlla i log di esecuzione di Lambda per capire perché il codice del test di convalida non riesce. Per informazioni sui log di esecuzione di Lambda, consulta Accessing Amazon logs for. CloudWatch AWS Lambda

L'ELB non può essere aggiornato a causa del seguente errore: Il gruppo target del taskset primario deve essere dietro l'ascoltatore

Problema: viene visualizzato il seguente messaggio di errore durante la distribuzione dell'applicazione Amazon ECS utilizzando: CodeDeploy

```
The ELB could not be updated due to the following error: Primary taskset target group must be behind listener
```

Possibile causa: questo errore può verificarsi se hai configurato un listener di test opzionale ed è configurato con un gruppo target sbagliato. Per ulteriori informazioni sul test listener in CodeDeploy, vedere Prima di iniziare una distribuzione Amazon ECS e. Cosa succede durante una distribuzione di Amazon ECS Per ulteriori informazioni sui set di attività, TaskSet consulta Amazon Elastic Container Service API Reference e describe-task-set nella sezione Amazon ECS del Command Reference. AWS CLI

Possibile soluzione: assicurati che il listener di produzione e il listener di test di Elastic Load Balancing puntino entrambi al gruppo target che attualmente serve i tuoi carichi di lavoro. Ci sono tre posti da controllare:

- In Amazon EC2, nelle impostazioni dei listener e delle regole del sistema di bilanciamento del carico. Per ulteriori informazioni, consulta Listeners for your Application Load Balancer nella User Guide for Application Load Balancers o Listeners for your Network Load Balancer nella User Guide for Network Load Balancer.
- In Amazon ECS, nel tuo cluster, nella configurazione di rete del tuo servizio. Per ulteriori informazioni, consulta le considerazioni su Application Load Balancer e Network Load Balancer nella Amazon Elastic Container Service Developer Guide.
- In CodeDeploy, nelle impostazioni del tuo gruppo di distribuzione. Per ulteriori informazioni, consulta Crea un gruppo di distribuzione per una distribuzione Amazon ECS (console).

La mia distribuzione a volte non riesce quando utilizzo Auto Scaling

Problema: stai utilizzando Auto Scaling con CodeDeploy e noti che le tue distribuzioni a volte falliscono. Per ulteriori informazioni sui sintomi di questo problema, consulta l'argomento intitolato Per i servizi configurati per utilizzare il service auto scaling and the blue/green deployment type, l'auto scaling non è bloccato durante una distribuzione, ma la distribuzione potrebbe non riuscire in alcune circostanze nella Amazon Elastic Container Service Developer Guide.

Possibile causa: questo problema può verificarsi in caso di conflitto tra i processi di Auto CodeDeploy Scaling.

Possibile soluzione: sospendi e riprendi i processi di Auto Scaling durante la CodeDeploy distribuzione utilizzando l'`RegisterScalableTarget`API (o il comando corrispondente). `register-scalable-target` AWS CLI Per ulteriori informazioni, consulta Sospendi e riprendi il ridimensionamento per Application Auto Scaling nella Guida per l'utente di Application Auto Scaling.

> **Note**
>
> CodeDeploy non posso chiamare direttamente. `RegisterScaleableTarget` Per utilizzare questa API, devi configurare CodeDeploy l'invio di una notifica o di un evento ad Amazon Simple Notification Service (o Amazon CloudWatch). È quindi necessario configurare Amazon SNS (or CloudWatch) per chiamare una funzione Lambda e configurare la funzione Lambda per chiamare l'API. `RegisterScalableTarget`

L'`RegisterScalableTarget`API deve essere chiamata con il `SuspendedState` parametro impostato su per sospendere le operazioni `true` di Auto Scaling e riprenderle. `false`
La notifica o l'evento CodeDeploy inviato deve avvenire all'avvio di una distribuzione (per attivare le operazioni di sospensione di Auto Scaling) o quando una distribuzione ha esito positivo, negativo o si interrompe (per attivare le operazioni di ripristino di Auto Scaling). Per informazioni su come configurare la generazione di notifiche o CodeDeploy eventi Amazon SNS, consulta. e. CloudWatch Monitoraggio delle distribuzioni con Amazon Events CloudWatch Monitoring Deployments with Amazon SNS Event Notifications

Solo ALB supporta il routing graduale del traffico, utilizza invece il routing AllAtOnce del traffico quando crei un gruppo di distribuzione create/update

Problema: viene visualizzato il seguente messaggio di errore durante la creazione o l'aggiornamento di un gruppo di distribuzione in: CodeDeploy

```
Only ALB supports gradual traffic routing, use AllAtOnce Traffic routing instead when you create/update Deployment group.
```

Possibile causa: questo errore può verificarsi se si utilizza un Network Load Balancer e si tenta di utilizzare una configurazione di distribuzione predefinita diversa da. `CodeDeployDefault.ECSAllAtOnce`

Possibili soluzioni.

- Modifica la configurazione di distribuzione predefinita in. `CodeDeployDefault.ECSAllAtOnce` Questa è l'unica configurazione di distribuzione predefinita supportata da Network Load Balancers.

  Per ulteriori informazioni sulle configurazioni di distribuzione predefinite, vedere. Configurazioni di distribuzione predefinite per una piattaforma di calcolo Amazon ECS
- Cambia il tuo load balancer in un Application Load Balancer. Application Load Balancer supporta tutte le configurazioni di distribuzione predefinite. Per ulteriori informazioni sulla creazione di un Application Load Balancer, vedere. Configura un sistema di bilanciamento del carico, gruppi target e listener per le distribuzioni di CodeDeploy Amazon ECS

Anche se la mia distribuzione è riuscita, il set di attività sostitutivo non supera i controlli di integrità di Elastic Load Balancing e la mia applicazione è inattiva

Problema: anche se CodeDeploy indica che la mia distribuzione è riuscita, il set di attività sostitutivo non supera i controlli di integrità di Elastic Load Balancing e la mia applicazione è inattiva.

Possibile causa: questo problema potrebbe verificarsi se hai eseguito una distribuzione CodeDeploy completa e il set di attività sostitutivo (verde) contiene un codice errato che causa il fallimento dei controlli di integrità di Elastic Load Balancing. Con la configurazione di distribuzione completa, i controlli di integrità del load balancer iniziano a essere eseguiti sul set di attività sostitutivo dopo che il traffico è stato trasferito su di esso (ovvero, dopo che si è verificato l'evento del ciclo di vita). CodeDeploy `AllowTraffic` Ecco perché i controlli sullo stato del set di attività sostitutive non andranno a buon fine dopo lo spostamento del traffico, ma non prima. Per informazioni sugli eventi del ciclo di vita che CodeDeploy vengono generati, consulta. Cosa succede durante una distribuzione di Amazon ECS

Possibili soluzioni.

- Modifica la configurazione di distribuzione da completa a canaria o lineare. In una configurazione canaria o lineare, i controlli di integrità del load balancer iniziano a essere eseguiti sul set di attività sostitutive durante l' CodeDeploy installazione dell'applicazione nell'ambiente di sostituzione e prima che il traffico venga spostato (ovvero durante l'evento del `Install` ciclo di vita e prima dell'evento). `AllowTraffic` Consentendo l'esecuzione dei controlli durante l'installazione dell'applicazione ma prima che il traffico venga spostato, verrà rilevato un codice errato dell'applicazione e causerà errori di distribuzione prima che l'applicazione diventi disponibile al pubblico.

  Per informazioni su come configurare le distribuzioni canarie o lineari, vedere. Modifica le impostazioni del gruppo di distribuzione con CodeDeploy

  Per informazioni sugli eventi del CodeDeploy ciclo di vita che si verificano durante una distribuzione di Amazon ECS, consulta. Cosa succede durante una distribuzione di Amazon ECS

  > **Note**
  >
  > Le configurazioni di distribuzione Canary e lineare sono supportate solo con Application Load Balancer.

- Se desideri mantenere la configurazione di distribuzione completa, configura un listener di test e controlla lo stato di integrità del task sostitutivo impostato con il lifecycle hook. `BeforeAllowTraffic` Per ulteriori informazioni, consulta Elenco degli hook relativi agli eventi del ciclo di vita per una distribuzione Amazon ECS.

Posso collegare più sistemi di bilanciamento del carico a un gruppo di distribuzione?

No. Se desideri utilizzare più Application Load Balancer o Network Load Balancer, utilizza gli aggiornamenti continui di Amazon ECS anziché le distribuzioni. CodeDeploy blue/green Per ulteriori informazioni sugli aggiornamenti continui, consulta Rolling update nella Amazon Elastic Container Service Developer Guide. Per ulteriori informazioni sull'utilizzo di più sistemi di bilanciamento del carico con Amazon ECS, consulta Registrazione di più gruppi target con un servizio nella Amazon Elastic Container Service Developer Guide.

Posso eseguire CodeDeploy blue/green implementazioni senza un sistema di bilanciamento del carico?

No, non è possibile eseguire CodeDeploy blue/green distribuzioni senza un load balancer. Se non riesci a utilizzare un sistema di bilanciamento del carico, utilizza invece la funzionalità di aggiornamenti continui di Amazon ECS. Per ulteriori informazioni sugli aggiornamenti continui di Amazon ECS, consulta Rolling update nella Amazon Elastic Container Service Developer Guide.

Come posso aggiornare il mio servizio Amazon ECS con nuove informazioni durante una distribuzione?

Per CodeDeploy aggiornare il servizio Amazon ECS con un nuovo parametro durante la distribuzione, specifica il parametro nella sezione del file. `resources` AppSpec Sono supportati solo alcuni parametri Amazon ECS CodeDeploy, ad esempio il file di definizione delle attività e i parametri del nome del contenitore. Per un elenco completo dei parametri di Amazon ECS che CodeDeploy possono essere aggiornati, consulta. AppSpec sezione «risorse» per le distribuzioni di Amazon ECS

> **Note**
>
> Se devi aggiornare il tuo servizio Amazon ECS con un parametro non supportato da CodeDeploy, completa queste attività:

1. Chiama l'`UpdateService`API di Amazon ECS con il parametro che desideri aggiornare. Per un elenco completo dei parametri che possono essere aggiornati, consulta UpdateService il riferimento alle API di Amazon Elastic Container Service.
2. Per applicare la modifica alle attività, crea una nuova blue/green distribuzione Amazon ECS. Per ulteriori informazioni, consulta Crea un'implementazione di Amazon ECS Compute Platform (console).

Risoluzione dei problemi AWS Problemi di implementazione Lambda

Argomenti

- AWS Lambda le distribuzioni falliscono dopo l'interruzione manuale di una distribuzione Lambda che non dispone di rollback configurati

AWS Lambda le distribuzioni falliscono dopo l'interruzione manuale di una distribuzione Lambda che non dispone di rollback configurati

In alcuni casi, l'alias di una funzione Lambda specificato in una distribuzione potrebbe fare riferimento a due diverse versioni della funzione. Il risultato è che i tentativi successivi di implementare la funzione Lambda falliscono. Una distribuzione Lambda può entrare in questo stato quando non ha i rollback configurati e viene interrotta manualmente. Per procedere, usa la AWS Lambda console per assicurarti che la funzione non sia configurata per spostare il traffico tra due versioni:

1. Accedi a Console di gestione AWS e apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Nel riquadro di sinistra scegliere Functions (Funzioni).
3. Seleziona il nome della funzione Lambda presente nella tua CodeDeploy distribuzione.
4. Da Alias, scegli l'alias utilizzato nella CodeDeploy distribuzione, quindi scegli Modifica.
5. Da Alias ponderato, scegli. **none** In questo modo si garantisce che l'alias non sia configurato per spostare una percentuale, o ponderazione, di traffico in più di una versione. Annotare la versione selezionata in Versione.
6. Scegli Save (Salva).

7. Apri la CodeDeploy console e prova a distribuire la versione visualizzata nel menu a discesa nel passaggio 5.

Risoluzione dei problemi relativi ai gruppi di distribuzione

Il tagging di un'istanza come parte di un gruppo di distribuzione non comporta automaticamente la distribuzione dell'applicazione nella nuova istanza

CodeDeploy non distribuisce automaticamente l'applicazione su un'istanza con nuovi tag. È necessario creare una nuova distribuzione nel gruppo di distribuzione.

Puoi utilizzarlo CodeDeploy per abilitare le distribuzioni automatiche su nuove istanze EC2 nei gruppi Amazon EC2 Auto Scaling. Per ulteriori informazioni, consulta Integrazione CodeDeploy con Amazon EC2 Auto Scaling.

Risoluzione dei problemi relativi alle istanze

Argomenti

- I tag devono essere impostati correttamente
- AWS CodeDeploy l'agente deve essere installato e in esecuzione sulle istanze
- Quando un'istanza viene terminata durante una distribuzione, la distribuzione continua senza errori per un'ora
- Analisi dei file di log per individuare gli errori di distribuzione nelle istanze
- Crea un nuovo file di CodeDeploy registro se è stato eliminato accidentalmente
- Risoluzione dei problemi relativi alla distribuzione «InvalidSignatureException — Firma scaduta: [ora] è precedente a [ora]»

I tag devono essere impostati correttamente

Utilizza il comando list-deployment-instances per confermare che le istanze utilizzate per una distribuzione siano etichettate correttamente. Se nell'output manca un'istanza EC2, utilizza la console EC2 per confermare che i tag sono stati impostati sull'istanza. Per ulteriori informazioni, consulta Lavorare con i tag nella console nella Guida per l'utente di Amazon EC2.

> **Note**
>
> Se tagghi un'istanza e la utilizzi immediatamente CodeDeploy per distribuirvi un'applicazione, l'istanza potrebbe non essere inclusa nella distribuzione. Questo perché possono essere necessari diversi minuti prima di CodeDeploy poter leggere i tag. Consigliamo di attendere almeno cinque minuti prima di provare a eseguire la distribuzione in un'istanza dopo averla contrassegnata.

AWS CodeDeploy l'agente deve essere installato e in esecuzione sulle istanze

Per verificare che l' CodeDeploy agente sia installato e in esecuzione su un'istanza, vedereVerifica che l' CodeDeploy agente sia in esecuzione.

Per installare, disinstallare o reinstallare l' CodeDeploy agente, consultaInstalla l' CodeDeploy agente.

Quando un'istanza viene terminata durante una distribuzione, la distribuzione continua senza errori per un'ora

CodeDeploy offre una finestra di un'ora per il completamento di ogni evento del ciclo di vita della distribuzione. Ciò consente ampio margine per l'esecuzione di script di lunga durata.

Se gli script non vengono eseguiti fino al completamento mentre è in corso un evento del ciclo di vita (ad esempio, se un'istanza viene terminata o l' CodeDeploy agente viene chiuso), potrebbe essere necessaria fino a un'ora prima che lo stato della distribuzione venga visualizzato come Non riuscito. Ciò accade anche se il periodo di timeout specificato nello script è inferiore a un'ora. Questo perché quando l'istanza viene terminata, l' CodeDeploy agente si spegne e non può elaborare altri script.

Se invece un'istanza viene terminata tra un evento e l'altro del ciclo di vita, o prima che inizi la prima fase dell'evento del ciclo di vita, il timeout si verifica dopo soli cinque minuti.

Analisi dei file di log per individuare gli errori di distribuzione nelle istanze

Se lo stato di un'istanza nella distribuzione è diverso da `Succeeded`, è possibile esaminare i dati del file di log di distribuzione per identificare il problema. Per informazioni su come accedere ai dati del log di distribuzione, consulta Visualizza i dati di registro per le CodeDeploy EC2/On-Premises distribuzioni.

Crea un nuovo file di CodeDeploy registro se è stato eliminato accidentalmente

Se si elimina accidentalmente il file di registro di distribuzione su un'istanza, CodeDeploy non crea un file di registro sostitutivo. Per creare un nuovo file di log, effettua l'accesso all'istanza ed esegui i comandi illustrati di seguito:

Per un'istanza Amazon Linux, Ubuntu Server o RHEL, esegui questi comandi in questo ordine, uno alla volta:

```
systemctl stop codedeploy-agent
```

```
systemctl start codedeploy-agent
```

Per un'istanza di Windows Server:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Risoluzione dei problemi relativi alla distribuzione «InvalidSignatureException — Firma scaduta: [ora] è precedente a [ora]»

CodeDeploy richiede riferimenti temporali accurati per eseguire le sue operazioni. Se la data e l'ora sull'istanza non sono impostate correttamente, potrebbero non corrispondere alla data di firma della richiesta di distribuzione, che viene CodeDeploy respinta.

Per evitare errori di distribuzione dovuti a impostazioni errate di data e ora, consulta i seguenti argomenti:

- Impostazione dell'orario di un'istanza Linux
- Impostazione dell'orario di un'istanza Windows

Risolvi i problemi relativi GitHub ai token

Token OAuth non valido GitHub

CodeDeploy le applicazioni create dopo giugno 2017 utilizzano token GitHub OAuth per ogni regione. AWS L'uso di token collegati a AWS regioni specifiche offre un maggiore controllo su quali CodeDeploy applicazioni hanno accesso a un repository. GitHub

Se ricevi un errore relativo al GitHub token, potresti avere un token più vecchio che ora non è valido.

Per correggere un token OAuth non valido GitHub

1. Rimuovi il vecchio token utilizzando uno dei seguenti metodi:
   - Per rimuovere il vecchio token utilizzando l'API, usa DeleteGitHubAccountToken.
   - Per rimuovere il vecchio token utilizzando AWS Command Line Interface:
     a. Vai al computer in cui si trova il token.
     b. Assicurati che AWS CLI sia installato su questo computer. Per le istruzioni di installazione, vedere Installazione, aggiornamento e disinstallazione di AWS CLI nella Guida per l'AWS Command Line Interface utente
     c. Immettete il seguente comando sul computer in cui risiede il token:

        ```
        aws delete-git-hub-account-token
        ```

        Per i dettagli sulla sintassi del comando, vedi delete-git-hub-account-token.
2. Aggiungi un nuovo token OAuth. Per ulteriori informazioni, consulta Integrazione con CodeDeploy GitHub.

È stato superato GitHub il numero massimo di token OAuth

Quando si crea una CodeDeploy distribuzione, il numero massimo di token consentiti è GitHub 10. Se ricevi un errore relativo ai token GitHub OAuth, assicurati di avere 10 token o meno. In presenza di più di 10 token, i primi token creati non sono validi. Ad esempio, se hai 11 token, il primo token creato non è valido. Se hai 12 token, i primi due token creati non sono validi. Per informazioni sull'utilizzo dell' CodeDeploy API per rimuovere i vecchi token, consulta. DeleteGitHubAccountToken

Risolvi i problemi relativi ad Amazon EC2 Auto Scaling

Argomenti

- Risoluzione dei problemi generali di Amazon EC2 Auto Scaling
- "CodeDeployRole non consente di eseguire le seguenti operazioni AWS service: AmazonAutoScaling "errore
- Le istanze in un gruppo Amazon EC2 Auto Scaling vengono continuamente fornite e terminate prima di poter implementare una revisione
- L'interruzione o il riavvio di un'istanza di Amazon EC2 Auto Scaling potrebbe causare il fallimento delle distribuzioni
- Evita di associare più gruppi di distribuzione a un singolo gruppo Amazon EC2 Auto Scaling
- Le istanze EC2 in un gruppo Amazon EC2 Auto Scaling non vengono avviate e ricevono l'errore «Heartbeat Timeout»
- I collegamenti non corrispondenti al ciclo di vita di Amazon EC2 Auto Scaling potrebbero causare l'interruzione o il fallimento delle distribuzioni automatiche nei gruppi di Amazon EC2 Auto Scaling
- Errore «La distribuzione non è riuscita perché non è stata trovata alcuna istanza per il tuo gruppo di distribuzione»

Risoluzione dei problemi generali di Amazon EC2 Auto Scaling

Le distribuzioni su istanze EC2 in un gruppo Amazon EC2 Auto Scaling possono fallire per i seguenti motivi:

- Amazon EC2 Auto Scaling avvia e chiude continuamente le istanze EC2. Se CodeDeploy non è possibile distribuire automaticamente la revisione dell'applicazione, Amazon EC2 Auto Scaling avvia e termina continuamente le istanze EC2.

  Dissocia il gruppo Amazon EC2 Auto Scaling dal gruppo di distribuzione o modifica CodeDeploy la configurazione del gruppo Amazon EC2 Auto Scaling in modo che il numero desiderato di istanze corrisponda al numero attuale di istanze (impedendo così ad Amazon EC2 Auto Scaling di lanciare altre istanze EC2). Per ulteriori informazioni, consulta Modifica le impostazioni del gruppo di distribuzione con CodeDeploy la sezione Manual Scaling for Amazon EC2 Auto Scaling.
- L' CodeDeploy agente non risponde. L' CodeDeploy agente potrebbe non essere installato se gli script di inizializzazione (ad esempio, gli script cloud-init) eseguiti immediatamente dopo l'avvio o

l'avvio di un'istanza EC2 impiegano più di un'ora per essere eseguiti. CodeDeploy dispone di un timeout di un'ora per consentire all'agente di rispondere alle distribuzioni in sospeso. CodeDeploy Per risolvere il problema, sposta gli script di inizializzazione nella revisione dell'applicazione CodeDeploy.

- Un'istanza EC2 in un gruppo Amazon EC2 Auto Scaling si riavvia durante una distribuzione. L'implementazione può fallire se un'istanza EC2 viene riavviata durante una distribuzione o se l' CodeDeploy agente viene spento durante l'elaborazione di un comando di distribuzione. Per ulteriori informazioni, consulta L'interruzione o il riavvio di un'istanza di Amazon EC2 Auto Scaling potrebbe causare il fallimento delle distribuzioni.
- Più revisioni delle applicazioni vengono distribuite contemporaneamente sulla stessa istanza EC2 in un gruppo Amazon EC2 Auto Scaling. La distribuzione contemporanea di più revisioni di applicazioni sulla stessa istanza EC2 in un gruppo Amazon EC2 Auto Scaling può fallire se una delle distribuzioni ha script che vengono eseguiti per più di qualche minuto. Non distribuire più revisioni di applicazioni sulle stesse istanze EC2 in un gruppo Amazon EC2 Auto Scaling.
- Una distribuzione non riesce per le nuove istanze EC2 lanciate come parte di un gruppo Amazon EC2 Auto Scaling. In questo scenario, l'esecuzione degli script in una distribuzione può impedire il lancio di istanze EC2 nel gruppo Amazon EC2 Auto Scaling. (È possibile che altre istanze EC2 nel gruppo Amazon EC2 Auto Scaling funzionino normalmente.) Per risolvere questo problema, assicurati che tutti gli altri script vengano completati per primi:
  - CodeDeploy l'agente non è incluso nell'AMI: se usi il cfn-init comando per installare l' CodeDeploy agente mentre avvii una nuova istanza, posiziona lo script di installazione dell'agente alla fine della `cfn-init` sezione del CloudFormation modello.
  - CodeDeploy l'agente è incluso nell'AMI: configura l'AMI in modo che l'agente si trovi in uno `Stopped` stato al momento della creazione dell'istanza, quindi includi uno script per l'avvio dell'agente come passaggio finale nella tua libreria di `cfn-init` script.

"CodeDeployRole non consente di eseguire le seguenti operazioni AWS service: AmazonAutoScaling "errore

Le distribuzioni che utilizzano un gruppo Auto Scaling creato con un modello di avvio richiedono le seguenti autorizzazioni. Queste si aggiungono alle autorizzazioni concesse dalla politica gestita. `AWSCodeDeployRole` AWS

- `EC2:RunInstances`
- `EC2:CreateTags`

- `iam:PassRole`

Potresti ricevere questo errore se ti mancano queste autorizzazioni. Per ulteriori informazioniTutorial: CodeDeploy Da utilizzare per distribuire un'applicazione in un gruppo di Auto Scaling, consulta Creazione di un modello di lancio per un gruppo di Auto Scaling e Autorizzazioni nella Guida per l'utente di Amazon EC2 Auto Scaling.

Le istanze in un gruppo Amazon EC2 Auto Scaling vengono continuamente fornite e terminate prima di poter implementare una revisione

In alcuni casi, un errore può impedire una corretta implementazione su nuove istanze di cui è stato effettuato il provisioning in un gruppo Amazon EC2 Auto Scaling. Ciò compromette la riuscita delle distribuzioni e l'integrità delle istanze. Poiché non è possibile eseguire o completare la distribuzione, le istanze vengono terminate subito dopo essere state create. La configurazione del gruppo Amazon EC2 Auto Scaling determina quindi il provisioning di un altro batch di istanze per cercare di soddisfare il requisito minimo di host integri. Anche questo batch viene terminato e il ciclo continua.

Tra le cause possibili sono incluse:

- Controlli di integrità del gruppo Amazon EC2 Auto Scaling non riusciti.
- Un errore nella revisione dell'applicazione.

Per risolvere il problema, esegui queste fasi:

1. Crea manualmente un'istanza EC2 che non fa parte del gruppo Amazon EC2 Auto Scaling. Contrassegna l'istanza con un tag di istanza EC2 univoco.
2. Aggiungi la nuova istanza al gruppo di distribuzione interessato.
3. Distribuisci una nuova revisione dell'applicazione senza errori nel gruppo di distribuzione.

Ciò richiede al gruppo Amazon EC2 Auto Scaling di distribuire la revisione dell'applicazione su istanze future nel gruppo Amazon EC2 Auto Scaling.

> **Note**
>
> Dopo aver confermato che le distribuzioni hanno avuto esito positivo, elimina l'istanza che hai creato per evitare addebiti continui sul tuo account. AWS

L'interruzione o il riavvio di un'istanza di Amazon EC2 Auto Scaling potrebbe causare il fallimento delle distribuzioni

Se un'istanza EC2 viene avviata tramite Amazon EC2 Auto Scaling e l'istanza viene quindi terminata o riavviata, le distribuzioni su quell'istanza potrebbero non riuscire per i seguenti motivi:

- Durante una distribuzione in corso, un evento di scalabilità in entrata o qualsiasi altro evento di terminazione causa il distacco dell'istanza dal gruppo Amazon EC2 Auto Scaling e quindi la sua chiusura. Poiché la distribuzione non può essere completata, ha esito negativo.
- L'istanza viene riavviata, ma l'avvio richiede più di cinque minuti. CodeDeploy considera questo come un timeout. Il servizio non è quindi in grado di completare nessuna distribuzione, corrente o futura, nell'istanza.

Per risolvere questo problema:

- In generale, accertati che tutte le distribuzioni siano state completate prima che l'istanza venga terminata o riavviata. Assicurati che tutte le distribuzioni vengano avviate dopo l'avvio o il riavvio dell'istanza.
- Le distribuzioni possono avere esito negativo se si specifica un'Amazon Machine Image (AMI) basata su Windows Server per una configurazione Amazon EC2 Auto Scaling e si utilizza il servizio EC2Config per impostare il nome del computer dell'istanza. Per risolvere questo problema, nell'AMI di base di Windows Server, nella scheda Generale delle proprietà del servizio EC2, deseleziona Imposta nome computer. Dopo aver deselezionato questa casella di controllo, questo comportamento viene disabilitato per tutte le nuove istanze di Windows Server Amazon EC2 Auto Scaling avviate con quell'AMI di base di Windows Server. Per le istanze Windows Server Amazon EC2 Auto Scaling su cui questo comportamento è abilitato, non è necessario deselezionare questa casella di controllo. È sufficiente ridistribuire nelle istanze le distribuzioni non riuscite dopo averle riavviate.

Evita di associare più gruppi di distribuzione a un singolo gruppo Amazon EC2 Auto Scaling

Come best practice, dovresti associare un solo gruppo di distribuzione a ciascun gruppo Amazon EC2 Auto Scaling.

Questo perché se Amazon EC2 Auto Scaling ridimensiona un'istanza con hook associati a più gruppi di distribuzione, invia notifiche per tutti gli hook contemporaneamente. In questo modo per ogni istanza vengono avviate più distribuzioni nello stesso momento. Quando più distribuzioni inviano comandi all' CodeDeploy agente contemporaneamente, è possibile che venga raggiunto il timeout di cinque minuti tra un evento del ciclo di vita e l'inizio della distribuzione o la fine dell'evento del ciclo di vita precedente. In questo caso, la distribuzione ha esito negativo, anche se un processo di distribuzione è in esecuzione come previsto.

> **Note**
>
> Il timeout predefinito per uno script in un evento del ciclo di vita è di 30 minuti. È possibile modificare il timeout impostando un valore diverso nel file. AppSpec Per ulteriori informazioni, consulta Aggiungi un AppSpec file per una distribuzione EC2/On-Premises.

Non è possibile controllare l'ordine con il quale le distribuzioni si verificano, se si prova a eseguire più distribuzioni contemporaneamente.

Infine, se la distribuzione su un'istanza non riesce, Amazon EC2 Auto Scaling interrompe immediatamente l'istanza. Quando la prima istanza si arresta, le altre distribuzioni in esecuzione iniziano a restituire errori. Poiché CodeDeploy prevede un timeout di un'ora prima che l' CodeDeploy agente risponda alle distribuzioni in sospeso, il timeout di ogni istanza può richiedere fino a 60 minuti.

Per ulteriori informazioni su Amazon EC2 Auto Scaling, consulta Under the CodeDeploy hood: and Auto Scaling integration.

Le istanze EC2 in un gruppo Amazon EC2 Auto Scaling non vengono avviate e ricevono l'errore «Heartbeat Timeout»

Un gruppo Amazon EC2 Auto Scaling potrebbe non riuscire a lanciare nuove istanze EC2, generando un messaggio simile al seguente:

`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.`

Il messaggio generalmente indica di procedere in uno dei seguenti modi:

- È stato raggiunto il numero massimo di implementazioni simultanee associate a un account. AWS Per ulteriori informazioni sui limiti di distribuzione, consulta CodeDeploy quote.

- Il gruppo Auto Scaling ha cercato di avviare troppe istanze EC2 troppo velocemente. Le chiamate API verso RecordLifecycleActionHeartbeato CompleteLifecycleActionper ogni nuova istanza sono state limitate.
- Un'applicazione in CodeDeploy è stata eliminata prima dell'aggiornamento o dell'eliminazione dei gruppi di distribuzione associati.

  Quando elimini un'applicazione o un gruppo di distribuzione, CodeDeploy tenta di ripulire tutti gli hook Amazon EC2 Auto Scaling ad esso associati, ma alcuni hook potrebbero rimanere. Se esegui un comando per eliminare un gruppo di distribuzione, gli hook rimanenti vengono restituiti nell'output. Tuttavia, se esegui un comando per eliminare un'applicazione, gli hook rimanenti non vengono visualizzati nell'output.

  Pertanto, come best practice, è opportuno eliminare tutti i gruppi di distribuzione associati a un'applicazione prima di eliminare l'applicazione. È possibile utilizzare l'output del comando per identificare gli hook del ciclo di vita che devono essere eliminati manualmente.

Se ricevi un messaggio di errore «Heartbeat Timeout», puoi determinare se i lifecycle hook rimanenti sono la causa e risolvere il problema procedendo come segue:

1. Esegui una delle seguenti operazioni:

   - Chiamate il comando delete-deployment-group per eliminare il gruppo di distribuzione associato al gruppo Auto Scaling che causa il timeout dell'heartbeat.
   - Chiamate il comando update-deployment-group con un elenco vuoto non nullo di nomi di gruppi Auto Scaling per scollegare tutti gli hook del ciclo di vita di Auto Scaling. CodeDeploy-managed

     Ad esempio AWS CLI , inserisci il seguente comando:

     ```
     aws deploy update-deployment-group --application-name my-example-app
     --current-deployment-group-name my-deployment-group --auto-scaling-
     groups
     ```

     Come altro esempio, se utilizzi l' CodeDeploy API con Java, chiama `UpdateDeploymentGroup` e imposta `autoScalingGroups` su`new ArrayList<String>()`. Questo imposta `autoScalingGroups` su un elenco vuoto e rimuove l'elenco esistente. Non usare`null`, che è l'impostazione predefinita, perché rimane così com'`autoScalingGroups`è, il che non è quello che vuoi.

Esaminare l'output della chiamata. Se l'output contiene una `hooksNotCleanedUp` struttura con un elenco di hook del ciclo di vita di Amazon EC2 Auto Scaling, ci sono degli hook del ciclo di vita rimanenti.

2. Chiama il comando describe-lifecycle-hooks, specificando il nome del gruppo Amazon EC2 Auto Scaling associato alle istanze EC2 che non sono state avviate. Nell'output, cerca uno dei seguenti elementi:

   - Nomi degli hook del ciclo di vita di Amazon EC2 Auto Scaling che corrispondono `hooksNotCleanedUp` alla struttura identificata nella fase 1.
   - Nomi degli hook del ciclo di vita di Amazon EC2 Auto Scaling che contengono il nome del gruppo di distribuzione associato al gruppo Auto Scaling che non funziona.
   - Nomi degli hook del ciclo di vita di Amazon EC2 Auto Scaling che potrebbero aver causato il timeout dell'heartbeat per la distribuzione. CodeDeploy

3. Se un hook rientra in una delle categorie elencate nel passaggio 2, chiama il comando delete-lifecycle-hook per eliminarlo. Specificare il gruppo Amazon EC2 Auto Scaling e l'hook del ciclo di vita nella chiamata.

   > **Important**
   >
   > Elimina solo gli hook che causano problemi, come indicato nel passaggio 2. Se elimini gli hook validi, le distribuzioni potrebbero fallire o CodeDeploy potrebbe non essere possibile distribuire le revisioni delle applicazioni su istanze EC2 con scalabilità orizzontale.

4. Chiama il comando update-deployment-group o create-deployment-group con i nomi dei gruppi Auto Scaling desiderati. CodeDeployreinstalla gli hook Auto Scaling con nuovi UUID.

> **Note**
>
> Se scolleghi un gruppo Auto Scaling da CodeDeploy un gruppo di distribuzione, tutte le distribuzioni in corso nel gruppo Auto Scaling potrebbero fallire e le nuove istanze EC2 scalate orizzontalmente dal gruppo Auto Scaling non riceveranno le revisioni dell'applicazione da. CodeDeploy Per far funzionare nuovamente Auto Scaling CodeDeploy, dovrai ricollegare

il gruppo Auto Scaling al gruppo di distribuzione e chiamarne uno nuovo per iniziare un'implementazione `CreateDeployment` a livello di flotta.

I collegamenti non corrispondenti al ciclo di vita di Amazon EC2 Auto Scaling potrebbero causare l'interruzione o il fallimento delle distribuzioni automatiche nei gruppi di Amazon EC2 Auto Scaling

Amazon EC2 Auto CodeDeploy Scaling utilizza gli hook del ciclo di vita per determinare quali revisioni delle applicazioni devono essere distribuite su quali istanze EC2 dopo il lancio nei gruppi di Amazon EC2 Auto Scaling. Le distribuzioni automatiche possono interrompersi o fallire se gli hook del ciclo di vita e le informazioni su questi hook non corrispondono esattamente in Amazon EC2 Auto Scaling e. CodeDeploy

Se le distribuzioni in un gruppo Amazon EC2 Auto Scaling falliscono, verifica se i nomi degli hook del ciclo di vita in Amazon EC2 Auto Scaling corrispondono. CodeDeploy In caso contrario, usa queste chiamate di comando. AWS CLI

Innanzitutto, ottieni l'elenco dei nomi degli hook del ciclo di vita sia per il gruppo Amazon EC2 Auto Scaling che per il gruppo di distribuzione:

1. Chiama il comando describe-lifecycle-hooks, specificando il nome del gruppo Amazon EC2 Auto Scaling associato al gruppo di distribuzione in. CodeDeploy Nell'output, nell'elenco `LifecycleHooks`, annotare ogni valore `LifecycleHookName`.
2. Chiama il comando get-deployment-group, specificando il nome del gruppo di distribuzione associato al gruppo Amazon EC2 Auto Scaling. Nell'output, nell'`autoScalingGroups`elenco, trova ogni elemento il cui valore del nome corrisponde al nome del gruppo Amazon EC2 Auto Scaling, quindi prendi nota del `hook` valore corrispondente.

Ora confrontare i due set di nomi di hook del ciclo di vita. Se corrispondono esattamente, carattere per carattere, allora non è il problema. Potresti provare altri passaggi per la risoluzione dei problemi di Amazon EC2 Auto Scaling descritti altrove in questa sezione.

Tuttavia, se i due set di nomi di hook del ciclo di vita non corrispondono esattamente, carattere per carattere, procedere nel seguente modo:

1. Se ci sono nomi di hook del ciclo di vita nell'output del comando describe-lifecycle-hooks che non sono anche nell'output del comando get-deployment-group, allora procedere nel seguente modo:

a. Per ogni nome di hook del ciclo di vita nell'output del comando describe-lifecycle-hooks, chiamare il comando delete-lifecycle-hook.

b. Chiama il comando update-deployment-group, specificando il nome del gruppo Amazon EC2 Auto Scaling originale. CodeDeploy crea nuovi hook del ciclo di vita sostitutivi nel gruppo Amazon EC2 Auto Scaling e associa i lifecycle hook al gruppo di distribuzione. Le distribuzioni automatiche dovrebbero ora riprendere non appena vengono aggiunte nuove istanze al gruppo Amazon EC2 Auto Scaling.

2. Se ci sono nomi di hook del ciclo di vita nell'output del comando get-deployment-group che non sono anche nell'output del comando describe-lifecycle-hooks, procedere nel seguente modo:

a. Chiama il comando update-deployment-group, ma non specifica il nome del gruppo Amazon EC2 Auto Scaling originale.

b. Richiama nuovamente il update-deployment-group comando, ma questa volta specifica il nome del gruppo Amazon EC2 Auto Scaling originale. CodeDeploy ricrea gli hook del ciclo di vita mancanti nel gruppo Amazon EC2 Auto Scaling. Le distribuzioni automatiche dovrebbero ora riprendere non appena vengono aggiunte nuove istanze al gruppo Amazon EC2 Auto Scaling.

Una volta che i due set di nomi degli hook del ciclo di vita corrispondono esattamente, carattere per carattere, le revisioni delle applicazioni devono essere nuovamente distribuite, ma solo su nuove istanze man mano che vengono aggiunte al gruppo Amazon EC2 Auto Scaling. Le distribuzioni non avvengono automaticamente su istanze già presenti nel gruppo Amazon EC2 Auto Scaling.

Errore «La distribuzione non è riuscita perché non è stata trovata alcuna istanza per il tuo gruppo di distribuzione»

Leggi questa sezione se vedi il seguente CodeDeploy errore:

```
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.
```

Le possibili cause di questo errore sono:

1. Le impostazioni del gruppo di distribuzione includono tag per le istanze EC2, le istanze locali o i gruppi di Auto Scaling che non sono corretti. Per risolvere questo problema, verifica che i tag siano corretti, quindi ridistribuisci l'applicazione.
2. La tua flotta si è ridimensionata dopo l'inizio dell'implementazione. In questo scenario, vengono visualizzate istanze integre nello `InService` stato del parco macchine, ma viene visualizzato anche l'errore riportato sopra. Per risolvere questo problema, ridistribuisci l'applicazione.
3. Il gruppo Auto Scaling non include alcuna istanza presente nello stato. `InService` In questo scenario, quando si tenta di eseguire una distribuzione su tutto il parco macchine, la distribuzione non riesce a causa del messaggio di errore riportato sopra perché è CodeDeploy necessaria almeno un'istanza nello stato. `InService` Esistono molte ragioni per cui potresti non avere istanze nello stato. `InService` Alcuni di essi includono:
   - Hai pianificato (o configurato manualmente) la dimensione del gruppo Auto Scaling in modo che sia. `0`
   - Auto Scaling ha rilevato istanze EC2 danneggiate (ad esempio, le istanze EC2 presentavano guasti hardware), quindi le ha annullate tutte, lasciandone nessuna nello stato. `InService`
   - Durante un evento di scalabilità orizzontale da `0` a`1`, CodeDeploy ha implementato una revisione precedentemente riuscita (chiamata ultima revisione riuscita) che era diventata inefficace dopo l'ultima implementazione. Ciò ha causato il fallimento della distribuzione sull'istanza con scalabilità orizzontale, che a sua volta ha causato l'annullamento dell'istanza da parte di Auto Scaling, senza lasciare alcuna istanza nello stato. `InService`

     Se scopri che non ci sono istanze nello `InService` stato, risolvi il problema come descritto nella procedura seguente,. To troubleshoot the error if there are no instances in the InService state

Per risolvere l'errore se non ci sono istanze nello stato InService

1. Nella console Amazon EC2, verifica l'impostazione della capacità desiderata. Se è zero, impostalo su un numero positivo. Attendi che l'istanza sia attiva`InService`, il che significa che la distribuzione è riuscita. Il problema è stato risolto e puoi saltare i passaggi rimanenti di questa procedura di risoluzione dei problemi. Per informazioni sull'impostazione della capacità desiderata, consulta Impostazione dei limiti di capacità sul gruppo Auto Scaling nella Amazon EC2 Auto Scaling User Guide.
2. Se Auto Scaling continua a tentare di lanciare nuove istanze EC2 per soddisfare la capacità desiderata ma non riesce mai a soddisfare lo scalabilità orizzontale, di solito ciò è dovuto a un guasto del ciclo di vita di Auto Scaling. Risolvi questo problema nel modo seguente:

a. Per verificare quale evento del ciclo di vita di Auto Scaling non va a buon fine, consulta Verifying a scaling activity for an Auto Scaling group nella Amazon EC2 Auto Scaling User Guide.

b. Se il nome dell'hook fallito è`CodeDeploy-managed-automatic-launch-deployment-hook-`*`DEPLOYMENT_GROUP_NAME`*, vai a CodeDeploy, trova il gruppo di distribuzione e trova la distribuzione fallita avviata da Auto Scaling. Quindi esamina il motivo per cui la distribuzione non è riuscita.

c. Se capisci perché la distribuzione non è riuscita (ad esempio, si sono verificati degli CloudWatch allarmi) e puoi risolvere il problema senza modificare la revisione, fallo ora.

d. Se, dopo un'indagine, stabilisci che CodeDeploy l'ultima revisione corretta non è più valida e non ci sono istanze integre nel tuo gruppo Auto Scaling, ti trovi in uno scenario di stallo di implementazione. Per risolvere questo problema, è necessario correggere la CodeDeploy revisione errata rimuovendo temporaneamente il lifecycle hook CodeDeploy del ciclo di vita dal gruppo Auto Scaling, quindi reinstallando l'hook e ridistribuendo una nuova (valida) revisione. Per istruzioni, consulta:

   - To fix the deployment deadlock issue (CLI)
   - To fix the deployment deadlock issue (console)

Per risolvere il problema del deadlock della distribuzione (CLI)

1. (Facoltativo) Blocca CI/CD le pipeline che causano l' CodeDeploy errore in modo che non si verifichino distribuzioni impreviste durante la risoluzione del problema.
2. Prendi nota della tua attuale impostazione Auto Scaling DesiredCapacity:

   `aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name` *`ASG_NAME`*

   Potrebbe essere necessario tornare a questo numero al termine di questa procedura.
3. Imposta l'DesiredCapacityimpostazione Auto Scaling su. 1 Questa opzione è facoltativa se la capacità desiderata era superiore 1 a quella iniziale. Riducendolo a`1`, l'istanza impiegherà meno tempo per il provisioning e la distribuzione in un secondo momento, il che velocizza la risoluzione dei problemi. Se la capacità desiderata di Auto Scaling era originariamente impostata su`0`, è necessario aumentarla a. 1 Questo è obbligatorio.

   aws autoscaling set-desired-capacity --auto-scaling-group-name --desired-capacity 1 *`ASG_NAME`*

> **Note**
>
> I passaggi rimanenti di questa procedura presuppongono che tu abbia impostato il tuo. DesiredCapacity1

A questo punto, Auto Scaling tenta di scalare fino a un'istanza. Quindi, poiché l'hook CodeDeploy aggiunto è ancora presente, CodeDeploy tenta di eseguire la distribuzione; la distribuzione fallisce; Auto Scaling annulla l'istanza e Auto Scaling tenta di riavviare un'istanza per raggiungere la capacità desiderata di una, operazione che ancora una volta fallisce. Sei in un ciclo di annullamento/riavvio.

4. De-register il gruppo Auto Scaling del gruppo di distribuzione:

> **Warning**
>
> Il comando seguente lancerà una nuova istanza EC2 senza software. Prima di eseguire il comando, assicuratevi che un'`InService`istanza di Auto Scaling che non esegue alcun software sia accettabile. Ad esempio, assicuratevi che il sistema di bilanciamento del carico associato all'istanza non invii traffico a questo host senza software.

> **Important**
>
> Utilizzate il CodeDeploy comando mostrato di seguito per rimuovere l'hook. Non rimuovere l'hook tramite il servizio Auto Scaling, poiché la rimozione non verrà riconosciuta da. CodeDeploy

```
aws deploy update-deployment-group --application-name APPLICATION_NAME
--current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-
groups
```

Dopo aver eseguito questo comando, si verifica quanto segue:

a. CodeDeploy annulla la registrazione del gruppo Auto Scaling dal gruppo di distribuzione.

b. CodeDeploy rimuove l'hook del ciclo di vita Auto Scaling dal gruppo Auto Scaling.

c. Poiché l'hook che stava causando una distribuzione fallita non è più presente, Auto Scaling annulla l'istanza EC2 esistente e ne lancia immediatamente una nuova per scalare fino alla capacità desiderata. La nuova istanza dovrebbe passare presto allo stato attuale. `InService` La nuova istanza non include software.

5. Attendi che l'istanza EC2 entri nello `InService` stato. Per verificarne lo stato, usa il seguente comando:

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names ASG_NAME --query AutoScalingGroups[0].Instances[*].LifecycleState
   ```

6. Aggiungi nuovamente l'hook all'istanza EC2:

   > **Important**
   >
   > Usa il CodeDeploy comando mostrato di seguito per aggiungere l'hook. Non utilizzare il servizio Auto Scaling per aggiungere l'hook, poiché l'aggiunta non verrà riconosciuta da. CodeDeploy

   ```
   aws deploy update-deployment-group --application-name APPLICATION_NAME --current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-groups ASG_NAME
   ```

   Dopo aver eseguito questo comando, si verifica quanto segue:

   a. CodeDeploy reinstalla l'hook del ciclo di vita Auto Scaling sull'istanza EC2

   b. CodeDeploy registra nuovamente il gruppo Auto Scaling con il gruppo di distribuzione.

7. Crea una distribuzione a livello di flotta con Amazon S3 GitHub o una versione che sai essere integra e che desideri utilizzare.

   Ad esempio, se la revisione è un file.zip in un bucket Amazon S3 `my-revision-bucket` chiamato con una chiave oggetto `httpd_app.zip` di, inserisci il seguente comando:

   ```
   aws deploy create-deployment --application-name APPLICATION_NAME --deployment-group-name DEPLOYMENT_GROUP_NAME --revision "revisionType=S3,s3Location={bucket=my-revision-bucket,bundleType=zip,key=httpd_app.zip}"
   ```

Poiché ora è presente un'`InService`istanza nel gruppo Auto Scaling, questa distribuzione dovrebbe funzionare e non dovresti più visualizzare l'errore La distribuzione non è riuscita perché non sono state trovate istanze per il tuo gruppo di distribuzione.

8. Una volta completata l'implementazione, ridimensionate il gruppo Auto Scaling riportandolo alla capacità originale, se in precedenza lo avevate ridimensionato in:

   ```
   aws autoscaling set-desired-capacity --auto-scaling-group-name ASG_NAME
   --desired-capacity ORIGINAL_CAPACITY
   ```

Per risolvere il problema del deadlock di distribuzione (console)

1. (Facoltativo) Blocca CI/CD le pipeline che causano l' CodeDeploy errore in modo che non si verifichino distribuzioni impreviste durante la risoluzione del problema.
2. Vai alla console Amazon EC2 e prendi nota dell'impostazione della capacità desiderata di Auto Scaling. Potrebbe essere necessario tornare a questo numero al termine di questa procedura. Per informazioni su come trovare questa impostazione, consulta Impostazione dei limiti di capacità nel gruppo Auto Scaling.
3. Imposta il numero desiderato di istanze EC2 su: 1

   Questo è facoltativo se la capacità desiderata era maggiore di 1 quella iniziale. Riducendolo a1, l'istanza impiegherà meno tempo per il provisioning e la distribuzione in un secondo momento, il che velocizza la risoluzione dei problemi. Se la capacità di Auto Scaling Desired era originariamente impostata su0, è necessario aumentarla a. 1 Questo è obbligatorio.

   > **Note**
   >
   > I passaggi rimanenti di questa procedura presuppongono che la capacità desiderata sia stata impostata su1.

   a. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/e scegli Auto Scaling Groups dal pannello di navigazione.

   b. Scegli la regione appropriata.

   c. Vai al gruppo Auto Scaling problematico.

   d. In Dettagli del gruppo, scegli Modifica.

e. Imposta la capacità desiderata su**1**.

f. Scegliere Aggiorna.

4. De-register il gruppo Auto Scaling del gruppo di distribuzione:

> **Warning**
>
> I seguenti passaggi secondari avvieranno una nuova istanza EC2 senza software. Prima di eseguire il comando, assicuratevi che un'`InService`istanza di Auto Scaling che non esegue alcun software sia accettabile. Ad esempio, assicuratevi che il sistema di bilanciamento del carico associato all'istanza non invii traffico a questo host senza software.

a. Apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy/.

b. Scegli la regione appropriata.

c. Nel riquadro di navigazione, scegliere Applications (Applicazioni).

d. Scegli il nome della tua CodeDeploy applicazione.

e. Scegli il nome del tuo gruppo CodeDeploy di distribuzione.

f. Scegli Modifica.

g. Nella configurazione dell'ambiente, deseleziona i gruppi Amazon EC2 Auto Scaling.

> **Note**
>
> La console non consente di salvare la configurazione se non è stata definita alcuna configurazione di ambiente. Per aggirare il controllo, aggiungi temporaneamente un tag di `EC2` o `On-premises` che sai non verrà risolto per nessun host. Per aggiungere un tag, seleziona istanze o On-premises istanze Amazon EC2 e aggiungi un tag Key di o. **`EC2` `On-premises`** Puoi lasciare il tag Value vuoto.

h. Scegli Save changes (Salva modifiche).

Dopo aver completato questi passaggi secondari, si verifica quanto segue:

i. CodeDeploy annulla la registrazione del gruppo Auto Scaling dal gruppo di distribuzione.

ii. CodeDeploy rimuove l'hook del ciclo di vita Auto Scaling dal gruppo Auto Scaling.

iii. Poiché l'hook che stava causando una distribuzione fallita non è più presente, Auto Scaling annulla l'istanza EC2 esistente e ne lancia immediatamente una nuova per scalare fino alla capacità desiderata. La nuova istanza dovrebbe passare presto allo stato attuale. `InService` La nuova istanza non include software.

5. Attendi che l'istanza EC2 entri nello `InService` stato. Per verificarne lo stato:

   a. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.

   b. Nel riquadro di navigazione, selezionare Groups (Gruppi Auto Scaling).

   c. Scegli il tuo gruppo Auto Scaling.

   d. Nel riquadro dei contenuti, scegli la scheda Gestione delle istanze.

   e. In Istanze, assicurati che la colonna Lifecycle indichi InServiceaccanto all'istanza.

6. Re-register il gruppo Auto Scaling con il gruppo di CodeDeploy distribuzione utilizzando lo stesso metodo utilizzato per rimuoverlo:

   a. Apri la CodeDeploy console all'indirizzo https://console.aws.amazon.com/codedeploy/.

   b. Scegli la regione appropriata.

   c. Nel riquadro di navigazione, scegliere Applications (Applicazioni).

   d. Scegli il nome della tua CodeDeploy applicazione.

   e. Scegli il nome del tuo gruppo CodeDeploy di distribuzione.

   f. Scegli Modifica.

   g. Nella configurazione dell'ambiente, seleziona i gruppi Amazon EC2 Auto Scaling e seleziona il tuo gruppo Auto Scaling dall'elenco.

   h. In Istanze o On-premisesistanze Amazon EC2, trova il tag che hai aggiunto e rimuovilo.

   i. Deseleziona la casella di controllo accanto alle istanze o alle istanze On-premises di Amazon EC2.

   j. Scegli Save changes (Salva modifiche).

   Questa configurazione reinstalla il lifecycle hook nel gruppo Auto Scaling.

7. Crea una distribuzione a livello di flotta con Amazon S3 GitHub o una versione che sai essere integra e che desideri utilizzare.

   Ad esempio, se la revisione è un file.zip in un bucket Amazon S3 `my-revision-bucket` chiamato con una chiave oggetto di, `httpd_app.zip` procedi come segue:

a. Nella CodeDeploy console, nella pagina Deployment Group, scegli Crea distribuzione.
b. Per Revision type (Tipo di revisione), scegliere My application is stored in Amazon S3 (La mia applicazione è archiviata in Amazon S3).
c. Per Posizione di revisione, scegli`s3://my-revision-bucket/httpd_app.zip`.
d. Per il tipo di file di revisione, scegli. `.zip`
e. Scegli Create deployment (Crea distribuzione).

Poiché ora è presente un'`InService`istanza nel gruppo Auto Scaling, questa distribuzione dovrebbe funzionare e non dovresti più visualizzare l'errore La distribuzione non è riuscita perché non sono state trovate istanze per il tuo gruppo di distribuzione.

8. Una volta completata l'implementazione, ridimensionate il gruppo Auto Scaling riportandolo alla capacità originale, se in precedenza lo avevate ridimensionato in:

a. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/e scegli Auto Scaling Groups dal pannello di navigazione.
b. Scegli la regione appropriata.
c. Vai al tuo gruppo Auto Scaling.
d. In Dettagli del gruppo, scegli Modifica.
e. Riporta la capacità desiderata al valore originale.
f. Scegliere Aggiorna.

Codici di errore per AWS CodeDeploy

Questo argomento fornisce informazioni di riferimento sugli CodeDeploy errori.

Codice di errore	Description
`AGENT_ISSUE`	La distribuzione non è riuscita a causa di un problema con l' CodeDeploy agente. Verifica che l'agente sia installato e in esecuzione su tutte le istanze in questo gruppo di distribuz ione. Ulteriori informazioni:

Codice di errore	Description
	• Verifica che l' CodeDeploy agente sia in esecuzione • Installa l' CodeDeploy agente • Collaborazione con l' CodeDeploy agente
AUTO_SCALING_IAM_ROLE_PERMISSIONS	Il ruolo di servizio associato al gruppo di distribuzione non dispone dell'autorizzazione richiesta per eseguire operazioni nel seguente AWS servizio. Ulteriori informazioni: • Fase 2: Creare un ruolo di servizio per CodeDeploy • Creazione di un ruolo per delegare le autorizzazioni a un servizio AWS
HEALTH_CONSTRAINTS	La distribuzione globale non è riuscita perché la distribuzione di troppe singole istanze non è riuscita, troppe poche istanze integre sono disponibili per la distribuzione, oppure alcune istanze nel gruppo di distribuzione stanno riscontrando dei problemi. Ulteriori informazioni: • Instance Health • Risoluzione dei problemi relativi alle istanze • Risolvi i problemi EC2/On-Premises di distribuzione

Codice di errore	Description
HEALTH_CONSTRAINTS_INVALID	La distribuzione non può iniziare perché il numero minimo di istanze integre, come definito dalla configurazione di distribuzione, non è disponibile. È possibile ridurre il numero di istanze integre richiesto aggiornando la configurazione della distribuzione o aumentando il numero di istanze in questo gruppo di distribuzione. Ulteriori informazioni: • Instance Health • Utilizzo delle istanze per CodeDeploy
IAM_ROLE_MISSING	La distribuzione non è riuscita perché non esiste alcun ruolo del servizio con nome del ruolo del servizio specificato per il gruppo di distribuzione. Verificare di utilizzare il nome del ruolo del servizio corretto. Ulteriori informazioni: • Fase 2: Creare un ruolo di servizio per CodeDeploy • Modifica le impostazioni del gruppo di distribuzione con CodeDeploy

Codice di errore	Description
`IAM_ROLE_PERMISSIONS`	CodeDeploy non dispone delle autorizzazioni necessarie per assumere un ruolo oppure il ruolo IAM che stai utilizzando non ti autorizza a eseguire operazioni in un servizio. AWS Ulteriori informazioni: • Fase 1: Configurazione • Fase 2: Creare un ruolo di servizio per CodeDeploy • Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2

Codice di errore	Description
NO_INSTANCES	Questo errore potrebbe essere causato da uno dei seguenti motivi. • Per una EC2/On-Premises blue/green distribuzione, se utilizzi i tag Amazon EC2, questi potrebbero non essere configurati correttamente. Nel tuo gruppo di CodeDeplo y distribuzione, assicurati che siano inclusi nelle istanze blu e nelle istanze verdi. Puoi confermare che le tue istanze sono etichetta te correttamente utilizzando la console Amazon EC2. • Se utilizzi un gruppo Amazon EC2 Auto Scaling, il gruppo Auto Scaling potrebbe non avere una capacità sufficiente. Assicurat i che il tuo gruppo Auto Scaling abbia una capacità sufficiente per l'implementazione. Puoi visualizzare la capacità del tuo gruppo Amazon EC2 Auto Scaling esaminando il numero di istanze integre utilizzando la console Amazon EC2. • Per una EC2/On-Premises blue/green distribuzione, le flotte blu e verde potrebbero non avere le stesse dimensioni. Verifica che i parchi istanze abbiano le stesse dimensioni. Ulteriori informazioni: • Tagging Instances for Deployments • Tutorial: CodeDeploy Da utilizzare per distribuire un'applicazione in un gruppo di Auto Scaling • Crea un'applicazione per una blue/green distribuzione (console)

Codice di errore	Description
OVER_MAX_INSTANCES	La distribuzione non è riuscita perché più istanze sono mirate per la distribuzione di quelle consentite per l'account. Per ridurre il numero di istanze mirate per questa distribuzione, aggiornare le impostazioni del tag per questo gruppo di distribuzione o eliminare alcune delle istanze mirate. In alternativa, puoi contattarci Supporto AWS per richiedere un aumento del limite. Ulteriori informazioni: • Modifica le impostazioni del gruppo di distribuzione con CodeDeploy • CodeDeploy quote • Richiedere un aumento del limite
THROTTLED	L'implementazione non è riuscita perché sono state fatte più richieste di quelle consentite AWS CodeDeploy da un ruolo IAM. Provare a ridurre il numero di richieste. Ulteriori informazioni: • Tasso di richiesta delle API query

Codice di errore	Description
UNABLE_TO_SEND_ASG	La distribuzione non è riuscita perché il gruppo di distribuzione non è configurato correttamente con il relativo gruppo Amazon EC2 Auto Scaling. Nella CodeDeploy console, elimina il gruppo Amazon EC2 Auto Scaling dal gruppo di distribuzione, quindi aggiungilo nuovamente. Ulteriori informazioni: • Under the Hood: CodeDeploy e l'integrazione con Auto Scaling

Argomenti correlati

Risoluzione dei problemi CodeDeploy

CodeDeploy risorse

Le seguenti risorse correlate possono aiutarti a lavorare con CodeDeploy.

Guide di riferimento e risorse di supporto

- AWS CodeDeploy Riferimento alle API: descrizioni, sintassi ed esempi di utilizzo su CodeDeploy azioni e tipi di dati, inclusi parametri e codici di errore comuni.
- CodeDeploy domande frequenti tecniche: le principali domande dei clienti in merito. CodeDeploy
- AWS centro assistenza: l'hub per la creazione e la gestione dei Supporto AWS casi. Include anche collegamenti ad altre risorse, come forum, domande frequenti tecniche, stato di salute del servizio e AWS Trusted Advisor.
- AWS piani di supporto: la pagina Web principale per informazioni sui Supporto AWS piani.
- Contattaci: un punto di contatto centrale per domande relative a AWS fatturazione, account, eventi, abusi e altre questioni.
- AWS termini del sito: informazioni dettagliate sul nostro copyright e marchio, sull'account, sulla licenza e sull'accesso al sito dell'utente e su altri argomenti.

Esempi

- CodeDeploy samples on GitHub: esempi e scenari di modelli per CodeDeploy.
- CodeDeploy Plugin Jenkins — Plugin Jenkins per. CodeDeploy
- CodeDeploy agente: Open-source versione dell'agente. CodeDeploy

Blog

- AWS DevOps blog — Approfondimenti per sviluppatori, amministratori di sistema e architetti.

AWS kit e strumenti di sviluppo software

I seguenti AWS SDK e strumenti supportano lo sviluppo di soluzioni con: CodeDeploy

- AWS SDK per .NET

- AWS SDK per Java
- AWS SDK per JavaScript
- AWS SDK per PHP
- AWS SDK per Python (Boto)
- AWS SDK per Ruby
- AWS Toolkit for Eclipse — Parti 1, 2 e 3.
- AWS Tools for Windows PowerShell— Un set di PowerShell cmdlet di Windows che espongono le funzionalità dell'ambiente AWS SDK per .NET . PowerShell
- CodeDeploy cmdlet in AWS Strumenti per PowerShell: set di PowerShell cmdlet di Windows che espongono le funzionalità dell'ambiente. CodeDeploy PowerShell
- AWS Command Line Interface— Una sintassi uniforme della riga di comando per l'accesso ai servizi. AWS AWS CLI Utilizza un unico processo di configurazione per consentire l'accesso a tutti i servizi supportati.
- AWS strumenti per sviluppatori: collegamenti a strumenti e risorse per sviluppatori che forniscono documentazione, esempi di codice, note di rilascio e altre informazioni per aiutarti a creare applicazioni innovative con CodeDeploy e AWS.

Cronologia dei documenti

La tabella riportata di seguito illustra le principali modifiche di questa guida per l'utente apportate per supportare le funzionalità nuove e avanzate aggiunte in seguito all'ultima versione della Guida per l'utente di CodeDeploy .

- Versione API: 2014-10-06

Modifica	Descrizione	Data
CodeDeploy versione agent v2.1.2 con supporto macOS	L' AWS CodeDeploy agente è stato aggiornato alla versione 2.1.2, che aggiunge il supporto per macOS sulle istanze Mac EC2. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente e Installazione dell' CodeDeploy agente per macOS.	13 ottobre 2026
CodeDeploy versione agent v2.1.1	L' AWS CodeDeploy agente è stato aggiornato alla versione 2.1.1. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente.	2 ottobre 2026
CodeDeploy versione agent v2.1.0	L' AWS CodeDeploy agente è stato aggiornato alla versione 2.1.0. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente.	7 settembre 2026
CodeDeploy versione agent v2.0.1	L' AWS CodeDeploy agente è stato aggiornato alla versione	24 agosto 2026

	2.0.1. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente.	
CodeDeploy versione agent v2.0.0	L' AWS CodeDeploy agente è stato aggiornato alla versione 2.0.0. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente.	15 luglio 2026
CodeDeploy versione agent v1.8.1	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.8.1. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente.	3 febbraio 2026
CodeDeploy versione agent v1.8.0	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.8.0. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente.	31 luglio 2025
CodeDeploy ha aggiornato una politica AWS gestita esistente	è `AWSCodeDeployDeployerAccess` stato aggiornato. Per ulteriori informazioni, consulta la sezione Aggiornamenti delle policy AWS gestite.	16 dicembre 2024
CodeDeploy ha aggiornato una politica AWS gestita esistente	è `AWSCodeDeployReadOnlyAccess` stato aggiornato. Per ulteriori informazioni, consulta la sezione Aggiornamenti delle policy AWS gestite.	16 dicembre 2024

CodeDeploy ha aggiornato una politica AWS gestita esistente	è `AWSCodeDeployFullAccess` stato aggiornato. Per ulteriori informazioni, consulta la sezione Aggiornamenti delle policy AWS gestite.	16 dicembre 2024
CodeDeploy versione agent v1.7.1	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.7.1. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente.	14 novembre 2024
Nomi di bucket Amazon S3 aggiornati	Sono stati aggiornati i bucket Amazon S3 di esempio in questa guida per utilizzare i nomi riservati da. AWS	17 giugno 2024
Aggiunto testo alternativo (alt text)	Tutte le immagini di questa guida sono state aggiornate per includere il testo alternativo. Il testo alternativo viene letto dagli screen reader e rende la nostra documentazione più accessibile agli utenti non vedenti.	22 maggio 2024
CodeDeploy versione agent v1.7.0	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.7.0. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente.	6 marzo 2024

Comandi modificati	I `sudo service codedeploy-agent` *`status|start|stop`* comandi non sono più consigliati perché non vengono utilizzati `systemd` per la gestione dei processi degli CodeDeploy agenti, che è una procedura consigliata. Per garantire che `systemd` venga utilizzato, utilizzar e `systemctl` i comandi, come mostrato nell'esempio seguente:`sytemctl start codedeploy-agent` . I seguenti argomenti sono stati aggiornati con `systemctl` i comandi: Installazione dell' CodeDeploy agente per Amazon Linux o RHEL, Installazione dell' CodeDeploy agente per Ubuntu Server, Risoluzione degli errori saltati di tutti gli eventi del ciclo di vita e Creazione di un nuovo file di CodeDeploy registro in caso di eliminazione accidentale.	12 gennaio 2024
Argomenti aggiunti	Sono stati aggiunti gli argomenti Gestione del processo CodeDeploy agente e Riferimento ai file negli script degli eventi del ciclo di vita.	12 gennaio 2024

CodeDeploy ora supporta le configurazioni zonali	È stato aggiornato l' CodeDeployargomento Crea una configurazione di distribuzione con informazioni sulla configurazione zonale.	7 dicembre 2023
CodeDeploy ora supporta le implementazioni di terminazione	È stato aggiunto l'argomento Abilitazione delle distribuzioni di terminazione durante gli eventi scale-in Auto Scaling per descrivere la funzionalità di distribuzione di terminazione. Sono state inoltre aggiornate la sezione AppSpec «hook» per una EC2/On-Premises distribuzione, gli argomenti Creazione di un gruppo di distribuzione per una distribuzione sul posto (console) e Creazione di un gruppo di distribuzione per una EC2/On-Premises blue/green distribuzione (console) per tenere conto della funzionalità.	7 dicembre 2023
Formattazione JSON fissa	È stata corretta la formattazione degli esempi di codice JSON nella sezione AppSpec «risorse» (solo Amazon ECS e AWS Lambda distribuzioni).	3 dicembre 2023
È stato aggiunto l'argomento sulla risoluzione dei problemi	È stato aggiunto un argomento sulla risoluzione dei problemi di distribuzione di Amazon ECS.	24 ottobre 2023

È stato aggiornato il nome del AppSpec file	È stato aggiornato il riferimento al CodeDeploy AppSpec file per indicare che il AppSpec file deve avere un nome `appspec.yml` per le EC2/On-Premises distribuzioni.	5 ottobre 2023
CodeDeploy ora supporta più sistemi di bilanciamento del carico	Sono stati aggiornati gli argomenti Crea un gruppo di distribuzione per una distribuzione locale (console), Crea un gruppo di distribuzione per una EC2/On-Premises blue/green distribuzione (console) e Configurazione di un bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2 per indicare il supporto per più sistemi di bilanciamento del carico.	26 settembre 2023
Aggiornato l'argomento Regioni nell'argomento VPC	È stata aggiornata la tabella nell'argomento Utilizzo CodeDeploy con Amazon Virtual Private Cloud per mostrare il supporto aggiuntivo per le regioni. In particolare, le regioni Asia Pacifico (Hyderabad), Asia Pacifico (Melbourne), Europa (Milano), Europa (Spagna) ed Europa (Zurigo) sono state aggiornate per mostrare il supporto per l'endpoint dell'agente.	22 settembre 2023

Regioni aggiornate nell'argomento del kit di risorse	Sono state aggiunte le seguenti AWS regioni ai nomi dei bucket del Resource kit per regione: Asia Pacifico (Osaka), Asia Pacifico (Hyderabad), Canada (Centrale), Europa (Spagna), Europa (Zurigo), Medio Oriente (Emirati Arabi Uniti). Inoltre, sono state aggiornate le politiche IAM con queste regioni e tutte le altre che mancavano.	22 settembre 2023
Argomenti abbreviati relativi all'installazione e all'aggiornamento degli agenti	Abbreviati gli argomenti Installazione dell' CodeDeploy agente per Windows Server e Aggiornamento dell' CodeDeploy agente su Windows Server. Sono stati rimossi gli URL ridondanti dei bucket Amazon S3 e i comandi di copia di Amazon S3.	21 settembre 2023
Aggiunta la regione Asia Pacifico (Giacarta)	È stata aggiunta l'area Asia Pacifico (Giacarta) ai nomi dei bucket del Resource Kit per regione.	21 settembre 2023
CodeDeploy ha aggiornato una politica gestita esistente AWS	La politica AWSCodeDeployRole gestita è stata aggiornata. Per ulteriori informazioni, consulta l'argomento relativo agli Aggiornamenti di AWS sulle policy gestite da AWS.	16 agosto 2023

È stato aggiunto un limite	È stato aggiunto un limite all'argomento CodeDeploy dei limiti. Il limite è il numero massimo di allarmi associati a un gruppo di distribuzione.	15 agosto 2023
È stato risolto un passaggio relativo ai sistemi di bilanciamento del carico	Sono state corrette le istruzioni in Creare un gruppo di distribuzione per una EC2/On-Premises blue/green distribuzione (console). La fase di bilanciamento del carico è ora contrassegnata come facoltativa.	3 agosto 2023
Formulazione chiarita nell'argomento Amazon ECS	È stata chiarita la formulazione in Tutorial: Deploy an application in Amazon ECS. La formulazione ora indica che stai distribuendo un'applicazione. In precedenza, il testo indicava che stavi distribuendo un servizio Amazon ECS.	3 agosto 2023
CodeDeploy è ora disponibile nella regione di Israele (Tel Aviv)	CodeDeploy è ora disponibile nella regione di Israele (Tel Aviv) (il-central-1). Diversi argomenti, inclusi quelli contenenti le istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione.	31 luglio 2023

Aggiornamento dell'argomento	È stato aggiornato l'argomento Risoluzione dei problemi di EC2/On-Premises distribuzione con un suggerimento sull'utilizzo di un runbook per eseguire automaticamente le attività di risoluzione dei problemi.	7 luglio 2023
Aggiornamento dell'argomento	È stata aggiornata la sezione «risorseAppSpec » per l'argomento sulle distribuzioni di Amazon ECS con ulteriori informazioni sulla definizione delle attività ARN.	7 luglio 2023
Aggiornamento dell'argomento	Aggiornamento del passaggio 1: installazione e configurazione dell'argomento AWS CLI sull'istanza locale con informazioni sulla risoluzione dei problemi.	7 luglio 2023
Aggiornamento dell'argomento	È stato aggiornato il Cross-serviceconfuso argomento sulla prevenzione dei deputati con informazioni sulle blue/green distribuzioni di Amazon ECS tramite. CloudFormation	06 luglio 2023
Aggiornamento dell'argomento	Il Cross-serviceconfuso argomento sulla prevenzione dei deputati è stato aggiornato con informazioni sulle blue/green distribuzioni di Amazon ECS tramite. CloudFormation	06 luglio 2023

Aggiornamento dell'argomento	È stato aggiornato l'argomento Configurazioni di distribuzione predefinite per una piattaforma di EC2/On-Premises elaborazione. È stata aggiunta una nota sul comportamento della configurazione di distribuzione `CodeDeployDefault.HalfAtATime` predefinita con i gruppi Auto Scaling.	29 giugno 2023
Aggiornamento dell'argomento	È stata aggiornata la sicurezza dell'infrastruttura nell' AWS CodeDeployargomento per indicare le nuove versioni minime e consigliate del protocollo Transport Layer Security (TLS).	28 giugno 2023
Limita l'aggiornamento	È stato modificato il seguente limite: «Numero massimo di ore di esecuzione EC2/On-Premises di una distribuzione sul posto». Per ulteriori informazioni, consulta i Limiti	27 giugno 2023
Aggiornamento dell'argomento	L'argomento Fase 3: Limitare le autorizzazioni CodeDeploy dell'utente è stato aggiornato con istruzioni dettagliate.	31 maggio 2023

CodeDeploy ha aggiornato una politica AWS gestita esistente	La politica AWSCodeDeployFullAccess gestita è stata aggiornata. Per ulteriori informazioni, consulta l'argomento relativo agli Aggiornamenti di AWS sulle policy gestite da AWS.	16 maggio 2023
CodeDeploy versione agent v1.6.0	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.6.0. Per ulteriori informazioni, vedere Cronologia delle versioni dell' CodeDeploy agente.	30 marzo 2023
CodeDeploy versione agent v1.5.0	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.5.0. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente.	3 marzo 2023
Aggiornamento della piattaforma di calcolo Amazon ECS	Le implementazioni su una piattaforma di elaborazione Amazon ECS sono ora supportate nella regione Asia Pacifico (Giacarta).	8 febbraio 2023
CodeDeploy ha aggiornato una politica gestita esistente AWS	La politica AWSCodeDeployRole gestita è stata aggiornata. Per ulteriori informazioni, consulta l'argomento relativo agli Aggiornamenti di AWS sulle policy gestite da AWS.	3 febbraio 2023

Aggiornamento dell'argomento	L'argomento Utilizzo CodeDeploy con Amazon Virtual Private Cloud è stato aggiornato con AWS regioni nuove e modificate.	2 febbraio 2023
Aggiornamenti dell'argomento	CodeDeploy è ora disponibi le nella regione Asia Pacifico (Melbourne) (ap-southeast-4). Diversi argomenti, inclusi quelli contenenti istruzion i per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di queste nuove regioni.	26 gennaio 2023
Aggiornamenti delle best practice di sicurezza	La CodeDeploy sezione Guida introduttiva e alcune altre sezioni sono state aggiornat e per renderle conformi alle migliori pratiche AWS di sicurezza.	23 gennaio 2023
CodeDeploy versione agent v1.4.1	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.4.1. Per ulteriori informazi oni, consulta Cronologia delle versioni dell' CodeDeploy agente.	6 dicembre 2022

È stato aggiunto l'argomento sulla risoluzione dei problemi	È stato aggiunto un argomento su come risolvere gli errori causati dai lunghi percorsi di file utilizzati con l' CodeDeploy agente per Windows. Per ulteriori informazioni, consulta I percorsi di file lunghi causano gli errori «Nessun file o directory di questo tipo».	6 dicembre 2022
È stato modificato un limite	È stato modificato il seguente limite: 'Numero massimo di configurazioni di distribuzione personalizzate associate a un AWS account'. Il limite ora è 200. Per ulteriori informazioni sui limiti, consulta l'argomento Limiti.	7 settembre 2022
CodeDeploy versione agent v1.4.0	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.4.0. Per ulteriori informazioni, vedere Cronologia delle versioni dell' CodeDeploy agente.	31 agosto 2022

Sono stati fissati alcuni limiti.	Sono stati fissati i seguenti limiti: «Il numero massimo di distribuzioni simultanee associate a un AWS account» è ora 1000. 'Il numero massimo di istanze in una singola implementazione' è ora 1000. 'Il numero massimo di istanze che possono essere utilizzate da distribuzioni simultanee in corso e associate a un account 'è ora 1000. «Il numero massimo di configurazioni di distribuzione personalizzate associate a un AWS account» è ora 100. Per ulteriori informazioni sui limiti, consulta l'argomento Limiti.	8 agosto 2022
È stata aggiunta una tabella che mostra gli CodeDeploy endpoint supportati in ciascuna regione.	Per ulteriori informazioni, consulta Utilizzo CodeDeploy con Amazon Virtual Private Cloud.	20 aprile 2022
È stato aggiunto un nuovo limite per le blue/green distribuzioni di Amazon ECS.	Il numero massimo di ore tra l'implementazione di una revisione e lo spostamento del traffico verso l'ambiente sostitutivo durante una blue/green distribuzione Amazon ECS è ora di 120. Per ulteriori informazioni, consulta Distribuzioni nell'argomento Limiti.	12 aprile 2022

È stato aggiunto un argomento su come prevenire il problema del vicesceriffo confuso	Per ulteriori informazioni, consulta AWS Identity and Access Management per AWS CodeDeploy.	14 marzo 2022
CodeDeploy ha aggiornato una politica AWS gestita esistente	Il `AmazonEC2RoleforAWSCodeDeployLimited` ruolo è stato aggiornato. Per ulteriori informazioni, consulta la sezione relativa agli aggiornamenti delle politiche AWS gestite.	22 novembre 2021
CodeDeploy ha aggiornato una politica AWS gestita esistente	è `AWS CodeDeployRole` stato aggiornato. Per ulteriori informazioni, consulta la sezione Aggiornamenti delle policy AWS gestite.	18 maggio 2021
CodeDeploy versione agent v1.3.2	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.3.2. Per ulteriori informazioni, vedere Cronologia delle versioni dell' CodeDeploy agente.	06 maggio 2021
CodeDeploy supporta l'aggiornamento di istanze Amazon EC2 obsolete	CodeDeploy ora supporta l'aggiornamento automatico delle istanze Amazon EC2 obsolete. Per ulteriori informazioni, consulta Configurare le opzioni avanzate per un gruppo di distribuzione.	23 febbraio 2021

CodeDeploy versione agent v1.3.1	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.3.1. Per ulteriori informazioni, vedere Cronologia delle versioni dell' CodeDeploy agente.	22 dicembre 2020
CodeDeploy versione agent v1.3.0	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.3.0. Per ulteriori informazioni, vedere Cronologia delle versioni dell' CodeDeploy agente.	10 novembre 2020
CodeDeploy versione agent v1.2.1	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.2.1. Per ulteriori informazioni, vedere Cronologia delle versioni dell' CodeDeploy agente.	23 settembre 2020
CodeDeploy supporta gli endpoint Amazon VPC con tecnologia AWS PrivateLink	Se utilizzi Amazon Virtual Private Cloud (Amazon VPC) per ospitare AWS le tue risorse, puoi stabilire una connessione privata tra il tuo VPC e. CodeDeploy È possibile utilizzare questa connessione per abilitare CodeDeploy per comunicare con le risorse nel VPC senza accedere all'Internet pubblico. Per ulteriori informazioni, consulta Utilizzo CodeDeploy con Amazon Virtual Private Cloud.	11 agosto 2020

Limiti CodeDeploy di servizio aggiornati	È stato aggiornato il limite per il numero di applicazioni per account e il numero di gruppi di distribuzione per applicazione a 1000. Per ulteriori informazioni sui limiti CodeDeploy del servizio, consulta CodeDeploy i limiti.	6 agosto 2020
CodeDeploy versione agent v1.1.2	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.1.2. Per ulteriori informazioni, vedere Cronologia delle versioni dell' CodeDeploy agente.	4 agosto 2020
CodeDeploy versione dell'agente 1.1.0 e integrazione con Amazon EC2 Systems Manager	La versione 1.1.0 dell' CodeDeploy agente è ora disponibile, per ulteriori informazioni, consulta Cronologia delle versioni dell'agente. CodeDeploy Ora puoi utilizzare Amazon EC2 Systems Manager per gestire CodeDeploy automaticamente l'installazione degli agenti e gli aggiornamenti sulle istanze Amazon EC2 o locali. Per ulteriori informazioni, consulta Installare l' CodeDeployagente utilizzando Amazon EC2 Systems Manager.	30 giugno 2020

CodeDeploy supporta la gestione delle blue/green distribuzioni Amazon ECS con CloudFormation	Ora puoi utilizzarlo CloudFormation per gestire le blue/green distribuzioni di Amazon ECS tramite. CodeDeploy La distribuzione viene generata definendo le risorse verdi e blu e specificando le impostazioni di routing e stabilizzazione del traffico da utilizzare in CloudFormation. Per ulteriori informazioni, consulta Creare una blue/green distribuzione Amazon ECS tramite CloudFormation.	19 maggio 2020
CodeDeploy supporta lo spostamento ponderato del traffico per le implementazioni Amazon ECS blue/green	CodeDeploy ora supporta lo spostamento ponderato del traffico per le implementazioni Amazon ECS blue/green. È possibile scegliere o creare una configurazione di distribuzione per specificare il numero di intervalli di instradamento del traffico nella distribuzione e la percentuale di traffico da instradare in ogni intervallo. Il seguente argomento è stato aggiornato per riflettere questa modifica: configurazioni di distribuzione su una piattaforma di calcolo Amazon ECS.	06 febbraio 2020

Argomenti aggiornati su sicurezza, autenticazione e controllo degli accessi	Le informazioni sulla sicurezza, l'autenticazione e il controllo degli accessi per sono CodeDeploy state organizzate in un nuovo capitolo sulla sicurezza. Per ulteriori informazioni, consulta Sicurezza.	26 novembre 2019
CodeDeploy supporta le regole di notifica	È ora possibile utilizzare le regole di notifica per notificare agli utenti le modifiche importanti nelle distribuzioni. Per ulteriori informazioni, consulta Creazione di una regola di notifica.	05 novembre 2019
Argomenti aggiornati	CodeDeploy è ora disponibile nella regione Asia Pacifico (Hong Kong) (ap-east-1). Diversi argomenti, inclusi quelli contenenti istruzioni per la configurazione dell'CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione. È necessario abilitare esplicitamente l'accesso a questa regione. Per ulteriori informazioni, vedere Gestione delle AWS regioni.	25 aprile 2019

Argomenti aggiornati	AWS CodeDeploy ora supporta blue/green le implementazioni di un'applic azione containerizzata in un servizio Amazon ECS. Un' CodeDeploy applicazione che utilizza la nuova piattaforma di calcolo Amazon ECS distribui sce un'applicazione container izzata per una nuova attività sostitutiva impostata nello stesso servizio Amazon ECS. Sono stati aggiunti e aggiornat i diversi argomenti per rifletter e questa modifica, tra cui Panoramica delle piattaforme di AWS CodeDeploy calcolo , Implementazioni su una piattaforma di calcolo Amazon ECS, AppSpec struttura dei file per le distribuzioni Amazon ECS e Creazione di un'applicazione per un'implem entazione di servizi Amazon ECS (Console).	27 novembre 2018
Agente aggiornato CodeDeplo y	L' AWS CodeDeploy agente è stato aggiornato alla versione 1.0.1.1597. Per ulteriori informazioni, consulta Cronologia delle versioni dell'agente. CodeDeploy	15 novembre 2018

Console aggiornata	Le procedure in questa guida sono state aggiornate per allinearle al nuovo design della console CodeDeploy.	30 ottobre 2018
Nuova versione minima supportata dell' CodeDeploy agente	La versione minima supportat a dell' AWS CodeDeploy agente è ora la 1.8.x. Per ulteriori informazioni, consulta Cronologia delle versioni dell' CodeDeploy agente.	7 agosto 2018

Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti apportate a ogni versione della Guida per l'utente di AWS CodeDeploy prima di giugno 2018.

Modifica	Descrizione	Data della modifica
Aggiornamenti di argomenti	CodeDeploy è ora disponibile nella regione Europa (Parigi) (eu-west-3). Diversi argomenti, inclusi quelli contenenti le istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione.	19 dicembre 2017
Argomenti aggiornat i	CodeDeploy è ora disponibile nella regione Cina (Ningxia). Per utilizzare i servizi nella regione Cina (Pechino) o nella regione Cina (Ningxia), è necessario disporre di un account e delle credenziali per tali aree. Gli account e le credenzia li di altre AWS regioni non funzionano per le regioni di Pechino e Ningxia e viceversa. Le informazioni su alcune risorse per le regioni della Cina, come i nomi dei bucket del CodeDeploy Resource Kit e le procedure di installazione degli CodeDeploy agenti, non	11 dicembre 2017

Modifica	Descrizione	Data della modifica
	sono incluse in questa edizione della Guida per l'CodeDeploy utente. Per ulteriori informazioni: • CodeDeployin Guida introduttiva AWS nella regione Cina (Pechino) • CodeDeploy Guida per l'utente per le regioni della Cina (versione inglese | versione cinese)	
Argomenti aggiornati	CodeDeploy ora supporta l'implementazione di una funzione Lambda. Una distribuzione AWS Lambda sposta il traffico in entrata da una funzione Lambda esistente a una versione della funzione Lambda aggiornata. È possibile scegliere o creare una configurazione di distribuzione per specificare il numero di intervalli di spostamento del traffico nella distribuzione e la percentuale di traffico da spostare in ciascun intervallo. AWS Lambda le distribuzioni sono supportate dal AWS Serverless Application Model (AWS SAM), quindi è possibile utilizzare una preferenza di distribuzione AWS SAM per gestire il modo in cui il traffico viene spostato durante un'implementazione. AWS Lambda Sono stati aggiunti e aggiornati diversi argomenti per riflettere questa modifica, inclusi Panoramica delle piattaforme di elaborazione CodeDeploy, Implementazioni su un AWS Piattaforma di elaborazione Lambda, Crea un AWS Implementazione della piattaforma di elaborazione Lambda (console), Crea un'applicazione per un AWS Lambda implementazione della funzione (console) e Aggiungi un AppSpec file per un AWS Implementazione Lambda.	28 novembre 2017

Modifica	Descrizione	Data della modifica
Nuovo argomento	CodeDeploy ora supporta le distribuzioni direttamente su un computer o un'istanza locale in cui è installato l'agente. CodeDeploy È possibile testare localmente una distribuzione e, in caso di errori, utilizzare i log degli errori degli CodeDeploy agenti per eseguirne il debug. È inoltre possibile utilizzare una distribuzione locale per verificare l'integrità di una revisione dell'applicazione, il contenuto di un AppSpec file e altro ancora. Per ulteriori informazioni, consulta Utilizzare l' CodeDeploy agente per convalidare un pacchetto di distribuzione su un computer locale.	16 novembre 2017
Argomenti aggiornati	CodeDeploy il supporto per i sistemi di bilanciamento del carico Elastic Load Balancing nei gruppi di distribuzione è stato ampliato per includere Network Load Balancer sia per le implementazioni che per le implementazioni in loco. blue/green Ora puoi scegliere un Application Load Balancer, Classic Load Balancer o Network Load Balancer per il tuo gruppo di distribuzione. I load balancer sono necessari per le implementazioni e opzionali per le blue/green implementazioni sul posto. Sono stati aggiunti e aggiornati i diversi argomenti per riflettere questo supporto, inclusi Integrating CodeDeploy with Elastic Load Balancing, Crea un'applicazione per un'implementazione sul posto (console), Prerequisiti di distribuzione, Integrating CodeDeploy with Elastic Load Balancing e Crea un'applicazione per un'implementazione sul posto (console).	12 settembre 2017

Modifica	Descrizione	Data della modifica
Argomenti aggiornati	CodeDeploy il supporto per i sistemi di bilanciamento del carico Elastic Load Balancing nei gruppi di distribuzione è stato ampliato per includere Application Load Balancer sia per le implementazioni che per le distribuzioni in loco. blue/green Ora puoi scegliere tra un Application Load Balancer e un Classic Load Balancer per il tuo gruppo di distribuzione. I load balancer sono necessari per le implementazioni e opzionali per blue/green le implement azioni sul posto. Gli argomenti che includono Integrating CodeDeploy with Elastic Load Balancing, Crea un'applic azione con CodeDeploy e Crea un gruppo di distribuzione con CodeDeploy sono stati aggiornati in modo da riflettere questo supporto aggiuntivo.	10 agosto 2017

Modifica	Descrizione	Data della modifica
Argomenti nuovi e aggiornati	CodeDeploy ora supporta l'utilizzo di più gruppi di tag per identificare unioni e intersezioni di istanze da includere in un gruppo di distribuzione. Se si utilizza un singolo gruppo di tag, qualsiasi istanza del gruppo identificata da almeno un tag viene inclusa nel gruppo di distribuzione. Se si utilizzano più gruppi di tag, vengono incluse solo le istanze che sono identificate da almeno un tag in ciascuno dei gruppi di tag. Per informazioni sul nuovo metodo per aggiungere istanze a un gruppo di distribuzione, consulta Tagging Instances for Deployments. Tra gli altri argomenti aggiornati per riflettere questo supporto sono inclusi Crea un'applicazione per un'implementazione sul posto (console), Crea un'applicazione per una blue/green distribuzione (console), Crea un gruppo di distribuzione per una distribuzione sul posto (console), Crea un gruppo di distribuzione per una distribuzione (console) EC2/On-Premises blue/green, Deployments e Passaggio 5: creare un'applicazione e un gruppo di distribuzione in Tutorial: utilizzare CodeDeploy per distribuire un'applicazione da GitHub.	31 luglio 2017
Argomento aggiornato	Sono stati aggiunti due metodi aggiuntivi per installare l' CodeDeploy agente sulle istanze di Windows Server. Installa l' CodeDeploy agente per Windows Server Oltre ai PowerShell comandi di Windows, sono ora disponibili istruzioni per scaricare il file di installazione utilizzando un collegamento HTTPS diretto e utilizzando un comando di copia Amazon S3. Dopo che il file è stato scaricato o copiato in un'istanza, è possibile eseguire l'installazione manualmente.	12 luglio 2017

Modifica	Descrizione	Data della modifica
Argomenti aggiornati	CodeDeploy ha migliorato il modo in cui gestisce le connessioni agli GitHub account e agli archivi. È ora possibile creare e archiviare fino a 25 connessioni agli GitHub account per associare CodeDeploy le applicazioni ai GitHub repository. Ogni connessione può supportare più repository. È possibile creare connessioni a un massimo di 25 GitHub account diversi o creare più di una connessione a un singolo account. Dopo aver collegato un'applicazione a un GitHub account, CodeDeploy gestisce le autorizzazioni di accesso richieste senza richiedere ulteriori azioni da parte dell'utente. Per riflettere questo supporto, sono stati aggiornati Specificare le informazioni su una revisione memorizzata in un repository GitHub, Integrazione con CodeDeploy GitHub e Tutorial: utilizzare CodeDeploy per distribuire un'applicazione da GitHub.	30 maggio 2017
Argomenti aggiornati	In passato, se l' CodeDeploy agente rilevava file in una posizione di destinazione che non facevano parte della revisione dell'applicazione relativa all'ultima distribuzione riuscita, per impostazione predefinita non riusciva a completare la distribuzione corrente. CodeDeploy ora offre opzioni su come l'agente gestisce questi file: non riuscire a completare la distribuzione, conservare il contenuto o sovrascriverlo. Crea una distribuzione con CodeDeployè stato aggiornato in base a questo supporto ed Comportamento di rollback con il contenuto esistente è stata aggiunta la nuova sezione. Ridistribuisci e ripristina una distribuzione con CodeDeploy	16 maggio 2017

Modifica	Descrizione	Data della modifica
Argomenti aggiornati	Un Classic Load Balancer in Elastic Load Balancing ora può essere assegnato a un gruppo di distribuzione utilizzando CodeDeploy la console o. AWS CLI Durante una distribuzione in locale, un sistema di bilanciamento del carico impedisce che il traffico Internet venga instradato a un'istanza mentre viene distribuito, quindi rende l'istanza nuovamente disponibile per il traffico una volta completata la distribuzione per quell'istanza. Sono stati aggiornati i diversi argomenti per riflettere questo nuovo supporto, tra cui Integrazione con altri AWS services, Integrating CodeDeploy with Elastic Load Balancing, Crea un'applicazione per un'implementazione sul posto (console), Crea un gruppo di distribuzione per una distribuzione sul posto (console) e AppSpec sezione 'hooks'. È stata aggiunta una nuova sezione alla guida alla risoluzione dei problemi, Risoluzione dei problemi relativi a un evento non riuscito o ApplicationStop AfterBlockTraffic relativo al BeforeBlockTraffic ciclo di vita dell'implementazione.	27 Aprile 2017

Modifica	Descrizione	Data della modifica
Argomenti aggiornati	Un Classic Load Balancer in Elastic Load Balancing ora può essere assegnato a un gruppo di distribuzione utilizzando CodeDeploy la console o. AWS CLI Durante una distribuzione in locale, un sistema di bilanciamento del carico impedisce che il traffico Internet venga instradato a un'istanza mentre viene distribuito, quindi rende l'istanza nuovamente disponibile per il traffico una volta completata la distribuzione per quell'istanza. Sono stati aggiornati i diversi argomenti per riflettere questo nuovo supporto, tra cui Integrazione con altri AWS services, Integrating CodeDeploy with Elastic Load Balancing, Crea un'applicazione per un'implementazione sul posto (console), Crea un gruppo di distribuzione per una distribuzione sul posto (console) e AppSpec sezione 'hooks'. È stata aggiunta una nuova sezione alla guida alla risoluzione dei problemi, Risoluzione dei problemi relativi a un evento non riuscito o ApplicationStop AfterBlockTraffic relativo al BeforeBlockTraffic ciclo di vita dell'implementazione.	1 maggio 2017

Modifica	Descrizione	Data della modifica
Argomenti aggiornati	CodeDeploy è ora disponibile nella regione Cina (Pechino). Per utilizzare i servizi nella regione Cina (Pechino) o nella regione Cina (Ningxia), è necessario disporre di un account e delle credenziali per tali aree. Gli account e le credenziali di altre AWS regioni non funzionano per le regioni di Pechino e Ningxia e viceversa. Le informazioni su alcune risorse per le regioni della Cina, come i nomi dei bucket del CodeDeploy Resource Kit e le procedure di installazione degli CodeDeploy agenti, non sono incluse in questa edizione della Guida per l'CodeDeploy utente. Per ulteriori informazioni: • CodeDeployin Guida introduttiva AWS nella regione Cina (Pechino) • CodeDeploy Guida per l'utente per le regioni della Cina (versione inglese | versione cinese)	29 marzo 2017

Modifica	Descrizione	Data della modifica
Argomenti nuovi e aggiornati	Sono stati introdotti diversi nuovi argomenti per riflettere il nuovo CodeDeploy supporto per le blue/green distribuzioni, un metodo di distribuzione in cui le istanze di un gruppo di distribuzione (l'ambiente originale) vengono sostituite da un diverso set di istanze (l'ambiente sostitutivo). Panoramica di una distribuzione blue/greenfornisce una spiegazione di alto livello della metodologia utilizzata da. blue/green CodeDeploy Nuovi argomenti aggiuntivi includono Crea un'applicazione per una blue/green distribuzione (console). Crea un gruppo di distribuzione per una distribuzione (console) EC2/On-Premises blue/green e Configura un sistema di bilanciamento del carico nelle distribuzioni Elastic Load Balancing for Amazon CodeDeploy EC2. Sono stati aggiornati numerosi argomenti, inclusi Crea una distribuzione con CodeDeploy, Utilizzo delle configurazioni di distribuzione in CodeDeploy, Crea un'applicazione con CodeDeploy, Lavorare con i gruppi di distribuzione in CodeDeploy , Utilizzo delle distribuzioni in CodeDeploy e AppSpec sezione 'hooks'.	25 gennaio 2017
Argomenti nuovi e aggiornati	Un nuovo argomento descrive come autenticare e registrare le istanze locali utilizzando credenziali temporanee aggiornate periodicamente generate tramite. Utilizza il comando register-on-premises-instance (ARN della sessione IAM) per registrare un'istanza locale Servizio di token di sicurezza AWS Questo approccio fornisce un supporto per parchi istanze locali di grandi dimensioni più valido rispetto al solo utilizzo delle credenziali di un utente IAM statico su ogni istanza. È stato aggiornato anche l'argomento Working with On-Premises Instances per riflettere il nuovo supporto.	28 dicembre 2016

Modifica	Descrizione	Data della modifica
Argomenti aggiornati	CodeDeploy è ora disponibile nella regione Europa (Londra) (eu-west-2). Diversi argomenti, inclusi quelli contenenti istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione.	13 dicembre 2016
Argomenti aggiornati	CodeDeploy è ora disponibile nella regione Canada (centrale) (ca-central-1). Diversi argomenti, inclusi quelli contenenti le istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione.	8 dicembre 2016
Argomenti aggiornati	CodeDeploy è ora disponibile nella regione Stati Uniti orientali (Ohio) (us-east-2). Diversi argomenti, inclusi quelli contenenti le istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione.	17 ottobre 2016
Nuovi argomenti	Una nuova sezione, Authentication and Access Control, fornisce informazioni complete sull'utilizzo di AWS Identity and Access Management (IAM) e CodeDeploy su come proteggere l'accesso alle risorse tramite l'uso di credenziali. Queste credenziali forniscono le autorizzazioni necessarie per accedere alle AWS risorse, ad esempio recuperare le revisioni delle applicazioni dai bucket Amazon S3 e leggere i tag sulle istanze Amazon EC2.	11 ottobre 2016

Modifica	Descrizione	Data della modifica
Argomento aggiornato	Aggiorna l' CodeDeploy agente su Windows Serverè stato aggiornato per riflettere la disponibilità del nuovo agent updater per Windows Server. CodeDeploy Se installato su un'istanza di Windows Server, il programma di aggiornamento verificherà periodicamente la presenza di nuove versioni. Quando rileva una nuova versione, la funzione di aggiornamento disinstalla la versione corrente dell'agente, se installata, prima di installare la versione più recente.	4 Ottobre 2016
Argomenti aggiornati	CodeDeploy ora si integra con gli CloudWatch allarmi di Amazon, rendendo possibile interrompere una distribuzione in caso di modifica dello stato di un allarme specificato per un numero di periodi consecutivi, come specificato nella soglia di allarme. CodeDeploy ora supporta anche il ripristino automatico di una distribuzione se vengono soddisfatte determinate condizioni, come un errore di distribuzione o un allarme attivato. Sono stati aggiornati diversi argomenti per riflettere queste modifiche, inclusi Crea un'applicazione con CodeDeploy, Crea un gruppo di distribuzione con CodeDeploy, Modifica le impostazioni del gruppo di distribuzione con CodeDeploy, Deployments, Ridistribuisci e ripristina una distribuzione con CodeDeploy e Integrazioni di prodotti e servizi con CodeDeploy, insieme a un nuovo argomento, Monitoraggio delle distribuzioni con allarmi in CloudWatch CodeDeploy.	15 settembre 2016

Modifica	Descrizione	Data della modifica
Argomenti nuovi e aggiornati	CodeDeploy ora fornisce l'integrazione con Amazon CloudWatch Events. Ora puoi utilizzare CloudWatch Events per avviare una o più azioni quando vengono rilevate modifiche nello stato di una distribuzione o nello stato di un'istanza che appartiene a un gruppo di CodeDeploy distribuzione. Puoi incorporare azioni che richiamano AWS Lambda funzioni, che pubblican o su Kinesis stream o argomenti di Amazon SNS, che inviano messaggi alle code di Amazon SQS o che, a loro volta, attivano azioni di allarme. CloudWatch Per ulteriori informazioni, consulta Monitoraggio delle distribuzioni con Amazon Events CloudWatch.	9 settembre 2016
Aggiornamenti di argomenti	Gli argomenti Integrating CodeDeploy with Elastic Load Balancing Integrazione con altri AWS services sono stati aggiornati per riflettere un'ulteriore opzione di bilanciam ento del carico. CodeDeploy ora supporta Classic Load Balancer e Application Load Balancer disponibili in Elastic Load Balancing.	11 agosto 2016
Aggiornamenti di argomenti	CodeDeploy è ora disponibile nella regione Asia Pacifico (Mumbai) (ap-south-1). Diversi argomenti, inclusi quelli contenenti istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione.	27 giugno 2016

Modifica	Descrizione	Data della modifica
Aggiornamenti di argomenti	CodeDeploy è ora disponibile nella regione Asia Pacifico (Seoul) (ap-northeast-2). Diversi argomenti, inclusi quelli contenenti istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione. Il sommario è stato riorganizzato per includere sezioni per istanze, configurazioni di distribuzione, applicazioni, gruppi di distribuzione, revisioni e distribuzioni. È stata aggiunta una nuova sezione per i tutorial CodeDeploy. Per una migliore fruibilità, diversi argomenti più lunghi, tra cui CodeDeploy AppSpec riferimento al file e Risoluzione dei problemi CodeDeploy, sono stati suddivisi in argomenti più brevi. Le informazioni di configurazione per l'agente CodeDeploy sono state spostate in un nuovo argomento, CodeDeploy riferimento alla configurazione dell'agente.	15 giugno 2016
Argomenti nuovi e aggiornati	Codici di errore per AWS CodeDeploy fornisce informazioni su alcuni dei messaggi di errore che potrebbero essere visualizzati quando le distribuzioni di CodeDeploy non vanno a buon fine. Le seguenti sezioni in Risoluzione dei problemi CodeDeploy sono state aggiornate per fornire una più valida assistenza nella risoluzione dei problemi di distribuzione: • Le istanze EC2 in un gruppo Amazon EC2 Auto Scaling non vengono avviate e ricevono l'errore «Heartbeat Timeout» • Evita di associare più gruppi di distribuzione a un singolo gruppo Amazon EC2 Auto Scaling	20 aprile 2016

Modifica	Descrizione	Data della modifica
Aggiornamenti di argomenti	CodeDeploy è ora disponibile nella regione Sud America (San Paolo) (sa-east-1). Diversi argomenti, inclusi quelli contenenti istruzioni per la configurazione dell' CodeDeplo yagente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione. Collaborazione con l' CodeDeploy agenteè stato aggiornat o in modo da riflettere la nuova opzione di configura zione:max_revisions:, che viene utilizzata per specifica re il numero di revisioni delle applicazioni per un gruppo di distribuzione che si desidera che l'agente archivi. CodeDeploy	10 marzo 2016

Modifica	Descrizione	Data della modifica
Argomenti nuovi e aggiornati	CodeDeploy ora supporta l'aggiunta di trigger a un gruppo di distribuzione per ricevere notifiche sugli eventi relativi alle distribuzioni o alle istanze in quel gruppo di distribuzione. Queste notifiche vengono inviate ai destinatari abbonati a un argomento di Amazon Simple Notification Service che hai inserito nell'azione del trigger. È inoltre possibile utilizzare i dati JSON creati quando viene attivato un trigger nel flusso di lavoro di notifica personalizzato. Per ulteriori informazioni, consulta Monitoring Deployments with Amazon SNS Event Notifications. Le procedure verranno aggiornate per riflettere la riprogettazione della pagina dei dettagli dell'applicazione. È stata aggiornata la sezione Quando un'istanza viene terminata durante una distribuzione, la distribuzione continua senza errori per un'ora in Risoluzione dei problemi CodeDeploy. CodeDeploy quoteè stato aggiornato in modo da riflettere i limiti rivisti relativi al numero di gruppi di distribuzione che possono essere associati a una singola applicazione, ai valori consentiti per le impostazioni relative al numero minimo di istanze integre e alle versioni richieste di. AWS SDK per Ruby	17 febbraio 2016

Modifica	Descrizione	Data della modifica
Argomenti nuovi e aggiornati	CodeDeploy è ora disponibile nella regione Stati Uniti occidentali (California settentrionale) (us-west-1). Diversi argomenti, inclusi quelli contenenti istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere l'aggiunta di questa nuova regione. In Scegli un tipo di CodeDeploy repository sono elencati e descritti i tipi di repository ora supportati da CodeDeploy. Questo nuovo argomento verrà aggiornato quando verrà introdotto il supporto per altri tipi di repository. Gestione delle operazioni degli CodeDeploy agentiè stato aggiornato con informazioni sul nuovo `.version` file aggiunto alle istanze per riportare la versione corrente dell' CodeDeploy agente, nonché informazioni sulle versioni supportate dell'agente. Alla guida per l'utente è stata aggiunta l'evidenziazione della sintassi per gli esempi di codice, inclusi esempi JSON e YAML. Aggiungere un file di specifiche dell'applicazione a una revisione per CodeDeploy è stato riorganizzato come istruzioni dettagliate.	20 gennaio 2016
Nuovo argomento	Distribuisci un'applicazione in un altro AWS accountdescrive i requisiti di configurazione e il processo per avviare le distribuzioni che appartengono a un altro account dell'organizzazione, senza bisogno di un set completo di credenziali per quell'altro account. Questa possibilità è molto utile per le organizzazioni che utilizzano più account per scopi differenti, ad esempio uno associato ad ambienti di sviluppo e test e un altro associato all'ambiente di produzione.	30 dicembre 2015

Modifica	Descrizione	Data della modifica
Aggiornamento di argomento	L'argomento Integrazioni di prodotti e servizi con CodeDeploy è stato riprogettato. Ora include una sezione per esempi di integrazione dalla community, con gli elenchi di post di blog ed esempi di video correlati a integrazioni CodeDeploy.	16 dicembre 2015
Aggiornamenti di argomenti	CodeDeploy è ora disponibile nella regione Asia Pacifico (Singapore) (ap-southeast-1). Diversi argomenti, inclusi quelli contenenti istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione.	9 dicembre 2015
Aggiornamenti di argomenti	Collaborazione con l' CodeDeploy agente è stato aggiornato per riflettere la nuova opzione `:proxy_uri:` nel file di configurazione dell'agente CodeDeploy. CodeDeploy AppSpec riferimento al file è stato aggiornato con informazioni sull'utilizzo di una nuova variabile di ambiente, `DEPLOYMENT_GROUP_ID`, a cui gli script hook possono accedere durante un evento del ciclo di vita della distribuzione.	1 dicembre 2015
Aggiornamento di argomento	Fase 2: Creare un ruolo di servizio per CodeDeployè stato aggiornato per riflettere una nuova procedura per la creazione di un ruolo di servizio CodeDeploy e per incorporare altri miglioramenti.	13 Novembre 2015
Aggiornamenti di argomenti	CodeDeploy è ora disponibile nella regione Europa (Francoforte) (eu-central-1). Diversi argomenti, inclusi quelli contenenti le istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione. L'argomento Risoluzione dei problemi CodeDeploy è stato aggiornato con informazioni su come verificare che le impostazioni di data e ora sulle istanze siano accurate.	19 ottobre 2015

Modifica	Descrizione	Data della modifica
Nuovi argomenti	CloudFormation modelli di CodeDeploy riferimentoè stato pubblicato per riflettere CloudFormation il nuovo sostegno CodeDeploy alle azioni. È stato creato un argomento Primary Components ed è stata introdotta una definizione di versione di destinazione.	1° ottobre 2015
Aggiornamenti di argomenti	Crea un gruppo di distribuzione con CodeDeploy è stato aggiornato per riflettere la possibilità di individuare le istanze per un gruppo di distribuzione utilizzando ricerche con caratteri jolly. Instance Health è stato aggiornato per chiarire il concetto di istanze integre minime.	31 agosto 2015
Aggiornamenti di argomenti	CodeDeploy è ora disponibile nella regione Asia Pacifico (Tokyo) (ap-northeast-1). Diversi argomenti, inclusi quelli contenenti istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di questa nuova regione.	19 agosto 2015
Aggiornamenti di argomenti	CodeDeploy ora supporta le implementazioni su istanze locali Red Hat Enterprise Linux (RHEL) e istanze Amazon EC2. Per ulteriori informazioni, consulta i seguenti argomenti: • Sistemi operativi supportati dall'agente CodeDeploy • Utilizzo delle istanze per CodeDeploy • Tutorial: Esegui WordPress la distribuzione su un'istanza Amazon EC2 (Amazon Linux o Red Hat Enterprise Linux e Linux, macOS o Unix) • Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux)	23 giugno 2015

Modifica	Descrizione	Data della modifica
Aggiornamento di argomento	CodeDeploy ora fornisce un set di variabili di ambiente che gli script di distribuzione possono utilizzare durante le distribuzioni. Queste variabili di ambiente includono informazioni come il nome dell' CodeDeploy applicazione corrente, il gruppo di distribuzione e l'evento del ciclo di vita della distribuzione, nonché l'identificatore di distribuzione corrente. CodeDeploy Per ulteriori informazioni, consulta la parte finale della sezione AppSpec sezione 'hooks' in CodeDeploy AppSpec riferimento al file.	29 maggio 2015
Aggiornamenti di argomenti	CodeDeploy ora fornisce una serie di policy AWS gestite in IAM che puoi utilizzare invece di creare manualmente le policy equivalenti da solo. Ciò include: • Una politica per consentire a un utente di registrar e le revisioni CodeDeploy solo con e poi distribuirle. CodeDeploy • Una politica per fornire a un utente l'accesso completo alle CodeDeploy risorse. • Una politica per fornire a un utente l'accesso in sola lettura alle CodeDeploy risorse. • Una policy da associare a un ruolo di servizio in modo da CodeDeploy poter identificare le istanze Amazon EC2 tramite i tag Amazon EC2, i tag delle istanze locali o i nomi dei gruppi Amazon EC2 Auto Scaling e distribuire le revisioni delle applicazioni di conseguenza. Per ulteriori informazioni, consulta la sezione Customer-managed esempi di politiche in Autenticazione e controllo degli accessi.	29 maggio 2015

Modifica	Descrizione	Data della modifica
Aggiornamenti di argomenti	CodeDeploy è ora disponibile nella regione Europa (Irlanda) (eu-west-1) e nella regione Asia Pacifico (Sydney) (ap-southeast-2). Diversi argomenti, inclusi quelli contenent i le istruzioni per la configurazione dell' CodeDeploy agente, sono stati aggiornati per riflettere la disponibilità di queste nuove regioni.	7 maggio 2015
Nuovi argomenti	CodeDeploy ora supporta le distribuzioni su istanze locali e su istanze Amazon EC2. I seguenti argomenti sono stati aggiunti per descrivere questo nuovo supporto: • Working with On-Premises Instances • Tutorial: Implementa un'applicazione su un'istanza locale con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux) • Working with On-Premises Instances	2 Aprile 2015
Nuovo argomento	È stato aggiunto l'argomento CodeDeploy risorse.	2 Aprile 2015

Modifica	Descrizione	Data della modifica
Aggiornamento di argomento	Risoluzione dei problemi CodeDeploy è stato aggiornato: • La nuova sezione Long-running i processi possono causare il fallimento delle distribuzioni descrive una procedura per identificare e risolvere gli errori di distribuzione dovuti a processi di lunga durata. • La Risoluzione dei problemi generali di Amazon EC2 Auto Scaling sezione è stata aggiornata per mostrare che CodeDeploy è stata aumentata la logica di timeout di Amazon EC2 Auto Scaling CodeDeploy per l'agente da cinque minuti a un'ora. • Una nuova I collegamenti non corrispondenti al ciclo di vita di Amazon EC2 Auto Scaling potrebbero causare l'interruzione o il fallimento delle distribuzioni automatiche nei gruppi di Amazon EC2 Auto Scaling sezione descrive i passaggi che puoi intraprendere per identificare e risolvere le distribuzioni automatiche fallite nei gruppi Amazon EC2 Auto Scaling.	2 Aprile 2015

Modifica	Descrizione	Data della modifica
Aggiornamenti di argomenti	I seguenti argomenti sono stati aggiornati per riflettere nuovi consigli per creare politiche personalizzate e quindi associarle a utenti e ruoli in IAM: • Configura un'istanza Amazon EC2 con cui lavorare CodeDeploy • Fase 4: crea un profilo di istanza IAM per le tue istanze Amazon EC2 • Fase 2: Creare un ruolo di servizio per CodeDeploy Sono state aggiunte due sezioni a Risoluzione dei problemi CodeDeploy: • Lista di controllo per la risoluzione di problemi generali • Per impostazione predefinita, PowerShell gli script di Windows non riescono a utilizzare la versione a 64 bit di Windows PowerShell La sezione AppSpec sezione 'hooks' di CodeDeploy AppSpec riferimento al file è stata aggiornata per descriver e in modo più preciso gli eventi del ciclo di vita di distribuz ione disponibili.	12 febbraio 2015
Aggiornamenti di argomenti	È stata aggiunta una nuova sezione a Risoluzione dei problemi CodeDeploy: Le istanze EC2 in un gruppo Amazon EC2 Auto Scaling non vengono avviate e ricevono l'errore «Heartbeat Timeout». È stata aggiunta una CloudBees sezione a. Integrazioni di prodotti e servizi con CodeDeploy	28 gennaio 2015

Modifica	Descrizione	Data della modifica
Aggiornamenti di argomenti	Sono state aggiunte le seguenti sezioni a Risoluzione dei problemi CodeDeploy: • L'utilizzo di alcuni editor di testo per creare AppSpec file e script di shell può causare il fallimento delle distribuzioni • L'utilizzo di Finder in macOS per includere in un file compresso la revisione di un'applicazione può determinare l'esito negativo delle distribuzioni • Risoluzione dei problemi relativi a un evento non riuscito o ApplicationStop AfterBlockTraffic relativo al BeforeBlockTraffic ciclo di vita dell'implementazione • Risoluzione dei problemi relativi a un evento del ciclo di vita di DownloadBundle distribuzione non riuscito con UnknownError: not open-for reading • Risoluzione dei problemi generali di Amazon EC2 Auto Scaling	20 gennaio 2015
Nuovi argomenti	La sezione Integrazioni di prodotti e servizi con CodeDeploy è stata aggiornata per includere i seguenti argomenti: • Integrazione CodeDeploy con Amazon EC2 Auto Scaling • Tutorial: CodeDeploy Da utilizzare per distribuire un'applicazione in un gruppo di Auto Scaling • Monitoring Deployments • Integrating CodeDeploy with Elastic Load Balancing • Integrazione con CodeDeploy GitHub • Tutorial: utilizzare CodeDeploy per distribuire un'applicazione da GitHub	9 gennaio 2015

Modifica	Descrizione	Data della modifica
Aggiornamenti di argomenti	• È stata aggiunta la sezione Implementa automaticamente da CodePipeline con CodeDeploy a Integrazione con CodeDeploy GitHub. Ora puoi attivare automaticamente una distribuzione da un GitHub repository ogni volta che il codice sorgente in quel repository viene modificato. • È stata aggiunta la sezione Risolvi i problemi relativi ad Amazon EC2 Auto Scaling a Risoluzione dei problemi CodeDeploy. Questa nuova sezione descrive come risolvere i problemi più comuni relativi alla distribuzione su gruppi Amazon EC2 Auto Scaling. • È stata aggiunta una nuova sottosezione di esempi di file alla sezione AppSpec sezione 'file' (solo distribuzioni) EC2/On-Premises di CodeDeploy AppSpec riferimento al file. Questa nuova sottosezione include diversi esempi di come utilizzare la `files` sezione di un AppSpec file CodeDeploy per indicare di copiare file o cartelle specifici in posizioni specifiche su un'istanza Amazon EC2 durante una distribuzione.	8 gennaio 2015
Nuovo argomento	Monitoring Deploymentsè stato aggiunto. CodeDeploy è integrato con AWS CloudTrail, un servizio che acquisisce le chiamate API effettuate da o per conto del CodeDeploy tuo AWS account e invia i file di registro a un bucket Amazon S3 da te specificato.	17 dicembre 2014
Versione pubblica iniziale	Si tratta della versione pubblica iniziale della Guida per l'utente di CodeDeploy .	12 novembre 2014

AWS Glossario

Per la AWS terminologia più recente, consultate il AWS glossario nella sezione Reference.Glossario AWS

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.