Guida per l'utente

AWS OpsWorks



Versione API 2013-02-18

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS OpsWorks: Guida per l'utente

Copyright © 2025 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 delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Che cos'è AWS OpsWorks? 1
AWS OpsWorks Servizi 1
AWS OpsWorks per Puppet Enterprise 4
Supporto regionale OpsWorks per Puppet Enterprise 5
Fine del ciclo di vita FAQs 6
In che modo i clienti esistenti saranno influenzati da questa fine del ciclo di vita? 6
Cosa succede ai miei server se non eseguo alcuna azione? 7
AWS OpsWorks for Puppet Enterprise Accetta nuovi clienti? 7
La fine del ciclo di vita influirà Regioni AWS su tutti allo stesso tempo? 7
Per quale livello di supporto tecnico è disponibile AWS OpsWorks for Puppet Enterprise? 7
Sono un cliente attuale di OpsWorks for Puppet Enterprise e ho bisogno di avviare un server in un account che in precedenza non utilizzava il servizio. Sono in grado di farlo? 7
Saranno disponibili nuove funzionalità per? AWS OpsWorks for Puppet Enterprise 8
Nozioni di base 8
Prerequisiti 8
Creare un master Puppet 13
Finire la configurazione 25
Aggiungere nodi da gestire 30
Effettuare l'accesso alla console Puppet Enterprise 33
Facoltativo: Usa CodeCommit 37
Crea un Puppet Master in CloudFormation 44
Prerequisiti 45
Creare un Puppet Enterprise Master in AWS CloudFormation 45
Aggiornare un server per utilizzare un dominio personalizzato 51
Prerequisiti 52
Limitazioni 52
Aggiornare un server per utilizzare un dominio personalizzato 53
Vedi anche 57
Lavorare con i tag 57
Come funzionano i tag in AWS OpsWorks for Puppet Enterprise 58
Aggiungi e gestisci tag in OpsWorks Puppet Enterprise (console) 59
Aggiungi e gestisci tag in OpsWorks Puppet Enterprise (CLI) 62
Vedi anche 67
Backup e ripristino di server 67

Esegui il backup di un OpsWorks Puppet Enterprise Server 68
Ripristina un file OpsWorks per Puppet Enterprise Server 72
Manutenzione sistema 73
Configurazione della manutenzione del sistema 75
Avvio della manutenzione del sistema on demand 77
Ripristino di configurazioni e file personalizzati dopo la manutenzione 78
Aggiunta automatica di nodi 78
Passaggio 1: crea un ruolo IAM da utilizzare come profilo di istanza 79
Fase 2. Creazione delle istanze tramite uno script di associazione automatica 79
Rimozione di nodi 81
Vedi anche 82
Eliminare un master Puppet 82
Passaggio 1: Disassociare i nodi gestiti 83
Passaggio 2: Eliminare il server 83
Vedi anche 83
Migrare un server Puppet su Amazon EC2 84
Passaggio 1: contatta Puppet per acquistare una licenza 84
Passaggio 2: Ottieni dettagli sul tuo server OpsWorks for Puppet Enterprise 84
Passaggio 3: Effettua un backup del tuo server OpsWorks for Puppet Enterprise 85
Fase 4: Avviare una nuova istanza EC2 86
Passaggio 5: installa Puppet Enterprise sulla nuova istanza EC2 87
Passaggio 6: ripristinare il backup sulla nuova EC2 istanza 88
Passaggio 7: configura la tua licenza Puppet 88
Fase 8: Migrazione dei nodi 89
Passaggio 9: Elimina il tuo server OpsWorks for Puppet Enterprise 91
Usando AWS CloudTrail 92
OpsWorks per informazioni su Puppet Enterprise in CloudTrail 92
Comprensione OpsWorks delle voci dei file di registro di Puppet Enterprise 93
Risoluzione dei problemi 95
Suggerimenti generali per la risoluzione dei problemi 96
Risoluzione di errori specifici 96
Supporto e informazioni della guida aggiuntivi 101
AWS OpsWorks per Chef Automate 102
Region Support per AWS OpsWorks for Chef Automate 105
Fine del ciclo di vita FAQs 106
In che modo gli utenti esistenti saranno influenzati da questa fine del ciclo di vita? 106

Cosa succede ai miei server se non eseguo alcuna azione? 107
A quali alternative posso passare? 107
Il servizio accetta ancora nuovi clienti? 107
La fine del ciclo di vita influirà Regioni AWS su tutti allo stesso tempo? 107
Che livello di supporto tecnico è disponibile? 107
Sono un cliente attuale di OpsWorks for Chef Automate e ho bisogno di avviare un server in un account che in precedenza non utilizzava il servizio. Sono in grado di farlo? 108
Ci saranno importanti novità nel corso del prossimo anno? 108
Aggiornamento a Chef Automate 2 108
Prerequisiti per l'aggiornamento a Chef Automate 2 108
Informazioni sul processo di aggiornamento 109
Aggiornamento a Chef Automate 2 (console) 109
Aggiornamento a Chef Automate 2 (CLI) 110
Ripristina un AWS OpsWorks for Chef Automate server su Chef Automate 1 (CLI) 111
Vedi anche 112
Nozioni di base 112
Prerequisiti 113
Creare un server Chef Automate 115
Terminare la configurazione e caricare i libri di ricette 129
Aggiungere nodi da gestire 138
Accedere al pannello di controllo di Chef Automate 145
Crea un server Chef Automate in CloudFormation 148
Prerequisiti 148
Creazione di un server Chef Automate in AWS CloudFormation 149
Aggiornare un server per utilizzare un dominio personalizzato 156
Prerequisiti 157
Limitazioni 52
Aggiornare un server per utilizzare un dominio personalizzato 53
Vedi anche 57
Rigenera lo starter kit 162
Rigenera lo AWS OpsWorks for Chef Automate starter kit con AWS CLI 163
Lavorare con i tag 164
Come funzionano i tag AWS OpsWorks for Chef Automate 165
Aggiungi e gestisci tag in AWS OpsWorks for Chef Automate (Console) 166
Aggiungi e gestisci tag in AWS OpsWorks for Chef Automate (CLI) 169
Vedi anche 174

Backup e ripristino di un server 174
Eseguire il backup di un server AWS OpsWorks for Chef Automate 175
Ripristina un AWS OpsWorks for Chef Automate server 177
Manutenzione sistema 179
Garantire che i nodi si fidino dell'Autorità di certificazione AWS OpsWorks 180
Configurazione della manutenzione del sistema 181
Avvio della manutenzione del sistema on demand 183
Ripristino di configurazioni e file personalizzati dopo la manutenzione 184
Scansioni di conformità 184
Conformità in Chef Automate 2.0 185
Confomità in Chef Automate 1.x 193
Aggiornamenti di conformità 199
Profili Compliance della community e personalizzati 199
Vedi anche 200
Rimozione di nodi 200
Argomenti correlati 201
Eliminazione di un server Chef Automate 201
Passaggio 1: Disassociare i nodi gestiti 202
Passaggio 2: Eliminare il server 202
Reimpostazione delle credenziali di Chef 203
Utilizzando AWS CloudTrail 204
AWS OpsWorks for Chef Automate Informazioni in CloudTrail 205
Comprensione delle AWS OpsWorks for Chef Automate voci dei file di registro 206
Risoluzione dei problemi 208
Suggerimenti generali per la risoluzione dei problemi 208
Risoluzione di errori specifici 209
Supporto e informazioni della guida aggiuntivi 216
Sicurezza nella gestione della AWS OpsWorks configurazione (CM) 218
Protezione dei dati 219
Integrazione con AWS Secrets Manager 220
Crittografia dei dati 221
Crittografia dei dati inattivi 221
Crittografia in transito 221
Gestione delle chiavi 221
Identity and Access Management 221
Destinatari 222

Autenticazione con identità ... 223
Gestione dell'accesso tramite policy ... 226
Come funziona AWS OpsWorks CM con IAM ... 229
Esempi di policy basate su identità ... 234
Risoluzione dei problemi ... 238
Policy gestite da AWS ... 240
Prevenzione intersettoriale confusa dei sostituti in AWS OpsWorks CM ... 248
Riservatezza del traffico Internet ... 252
Registrazione e monitoraggio ... 252
Convalida della conformità ... 252
Resilienza ... 253
Sicurezza dell'infrastruttura ... 254
Analisi della configurazione e delle vulnerabilità ... 255
Best practice di sicurezza ... 255
AWS OpsWorks Pile ... 257
Stack ... 260
Livelli ... 260
Ricette ed eventi LifeCycle ... 260
Istanze ... 261
App ... 262
Personalizzazione dello stack ... 263
Gestione delle risorse ... 264
Sicurezza e autorizzazioni ... 264
Monitoraggio e registrazione ... 264
CLI, SDK e modelli AWS CloudFormation ... 265
Fine del ciclo di vita FAQs ... 266
In che modo i clienti esistenti saranno influenzati da questa fine del ciclo di vita? ... 266
AWS OpsWorks Stacks Accetta nuovi clienti? ... 266
Dove devo migrare i miei stack esistenti? ... 266
Come posso mantenere le mie EC2 istanze Amazon esistenti dopo la fine del ciclo di vita? ... 267
La fine del ciclo di vita influirà su tutti Regioni AWS allo stesso tempo? ... 267
Per quale livello di supporto tecnico è disponibile AWS OpsWorks Stacks? ... 267
Saranno rilasciate nuove funzionalità per? AWS OpsWorks Stacks ... 268
Migrazione delle applicazioni a Systems Manager Application Manager ... 268
Come funziona lo script ... 269

Prerequisiti 269
Limitazioni 270
Nozioni di base 270
Domande frequenti 286
Risoluzione dei problemi 298
Utilizzo dello strumento AWS OpsWorks Stacks Detach in Place 299
Come funziona il processo 300
Limitazioni 302
Nozioni di base 303
Nozioni di base 312
Supporto nelle regioni 313
Nozioni di base: esempio 314
Nozioni di base: Linux 336
Nozioni di base: Windows 368
Nozioni di base: libri di ricette 403
Best practice 439
Storage del dispositivo root 439
Ottimizzazione del numero di server 442
Gestione delle autorizzazioni 445
Gestione e distribuzione di app e libri di ricette 448
Creazione di pacchetti di dipendenze di libri di ricette a livello locale 458
Stack 463
Migra gli stack da -Classic EC2 464
Creare un nuovo stack 467
Esecuzione di uno stack in un VPC 476
Aggiornamento di uno stack 488
Clonazione di uno stack 489
Esecuzione dei comandi dello stack 491
Utilizzo di un JSON personalizzato 494
Eliminazione di uno stack 497
Livelli 501
OpsWorks Nozioni di base sui livelli 502
Elastic Load Balancing Layer 518
Amazon RDS Service Layer 523
Livelli del cluster ECS 529
Livelli personalizzati 536

Installazioni pacchetto per livello ... 537
Istanze ... 538
Utilizzo delle AWS OpsWorks istanze Stacks ... 539
Utilizzo di risorse informatiche create al di fuori di AWS OpsWorks Stacks ... 601
Modifica della configurazione dell'istanza ... 649
Eliminazione delle AWS OpsWorks istanze di Stacks ... 651
Accesso con SSH ... 653
Accesso con RDP ... 656
App ... 661
Aggiunta di app ... 662
Distribuzione di app ... 669
Modifica delle app ... 674
Connessione a un database ... 675
Utilizzo delle variabili di ambiente ... 677
Passaggio di dati alle applicazioni ... 679
Utilizzo di chiavi SSH di repository Git ... 683
Utilizzo di domini personalizzati ... 684
Uso di SSL ... 686
Libri di ricette e ricette ... 694
Repository dei libri di ricette ... 695
Versione di Chef ... 699
Versioni di Ruby ... 718
Installazione di libri di ricette personalizzati ... 720
Aggiornamento dei libri di ricette personalizzati ... 724
Esecuzione di ricette ... 726
Gestione delle risorse ... 735
Registrazione di risorse con uno stack ... 737
Collegamento e spostamento di risorse ... 742
Scollegamento delle risorse ... 748
Annullamento della registrazione delle risorse ... 751
Tag ... 753
Impostazione di tag a livello di stack ... 754
Impostazione di tag a livello di livello ... 756
Gestione dei tag con AWS CLI ... 758
Limitazioni dei tag ... 759
Monitoraggio ... 760

Usare Amazon CloudWatch ... 760
Usando AWS CloudTrail ... 773
Utilizzo di Amazon CloudWatch Logs ... 776
Utilizzo di Amazon CloudWatch Events ... 781
Sicurezza e autorizzazioni ... 782
Gestione delle autorizzazioni utente ... 784
Consentire a AWS OpsWorks Stacks di agire per tuo conto ... 809
Prevenzione del "confused deputy" ... 814
Specificazione delle autorizzazioni per le app in esecuzione su istanze EC2 ... 818
Gestione dell'accesso SSH ... 822
Gestione degli aggiornamenti della sicurezza ... 830
Utilizzo dei gruppi di sicurezza ... 832
Chef 12 Linux ... 835
Panoramica ... 836
Passaggio a Chef 12 ... 837
Sistemi operativi supportati ... 838
Tipi di istanze supportati ... 838
Ulteriori informazioni ... 838
Spostamento ai contenitori di dati ... 839
Versioni precedenti di Chef ... 841
Chef 11.10 e versioni precedenti per Linux ... 841
Utilizzo di AWS OpsWorks Stacks con altri servizi AWS ... 1282
Utilizzo di un archivio dati di back-end ... 1283
ElastiCache Redis ... 1292
Utilizzo di un bucket Amazon S3 ... 1306
Utilizzo AWS CodePipeline con AWS OpsWorks Stacks ... 1321
Utilizzo della CLI AWS OpsWorks di Stacks ... 1385
Creazione di un'istanza ... 1387
Distribuire un'app ... 1390
Elencare app ... 1392
Comandi di elenco ... 1393
Elencare distribuzioni ... 1394
Elencare indirizzi IP elastici ... 1396
Elencare istanze ... 1397
Elencare stack ... 1398
Elencare livelli ... 1400

Eseguire una ricetta 1404
Install Dependencies (Installa dipendenze) 1405
Aggiornare la configurazione dello stack 1406
Guida al debugging e alla risoluzione dei problemi 1407
Il debug delle ricette 1408
Risoluzione dei problemi e debug di alcuni problemi comuni 1426
AWS OpsWorks CLI di Stacks Agent 1436
agent_report 1439
get_json 1439
instance_report 1444
list_commands 1445
run_command 1446
show_log 1447
stack_state 1448
AWS OpsWorks Riferimento a Stacks Data Bag 1451
Contenitore di dati di app (aws_opsworks_app) 1455
Contenitore di dati di comando (aws_opsworks_command) 1459
Borsa dati del cluster Amazon ECS (aws_opsworks_ecs_cluster) 1461
Borsa dati Elastic Load Balancing (aws_opsworks_elastic_load_balancer) 1462
Contenitore di dati dell'istanza (aws_opsworks_instance) 1463
Contenitori di dati di livello (aws_opsworks_layer) 1468
Borsa dati Amazon RDS (aws_opsworks_rds_db_instance) 1471
Contenitore di dati dello stack (aws_opsworks_stack) 1472
Contenitore di dati utente (aws_opsworks_user) 1474
OpsWorks Modifiche agli agenti 1476
Rilasci dell'agente Chef 12 1476
Rilasci dell'agente Chef 11.10 1480
Risorse 1486
Guide di riferimento, strumenti e risorse di supporto 1486
AWS Kit di sviluppo software 1487
Software open source 1488
AWS OpsWorks cronologia dei documenti 1489
Aggiornamenti precedenti 1497
..... mdi

Che cos'è AWS OpsWorks?

> **Important**
>
> I AWS OpsWorks servizi hanno raggiunto la fine del loro ciclo di vita e sono stati disattivati sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks è un servizio di gestione della configurazione che consente di configurare e gestire le applicazioni in un'azienda cloud utilizzando Puppet o Chef. AWS OpsWorks Impila e ti AWS OpsWorks for Chef Automate consente di utilizzare i libri di cucina e le soluzioni Chef per la gestione della configurazione, mentre OpsWorks per Puppet Enterprise ti consente di configurare un server master Puppet Enterprise in. AWS Puppet offre una serie di strumenti per applicare lo stato desiderato della tua infrastruttura e automatizzare le attività on demand.

AWS OpsWorks Servizi

AWS OpsWorks per Puppet Enterprise

OpsWorks per Puppet Enterprise ti consente di creare server AWS master Puppet gestiti. Un server master Puppet gestisce i nodi nella tua infrastruttura, archivia dati su questi nodi e funge da repository centrale per i moduli Puppet. I moduli sono unità riutilizzabili e condivisibili di codice Puppet contenenti le istruzioni su come configurare la tua infrastruttura. Puoi scaricare i moduli dalla community da Puppet Forge oppure utilizzare il kit di sviluppo Puppet per creare i tuoi moduli personalizzati, quindi gestirne la distribuzione con Puppet Code Manager.

OpsWorks for Puppet Enterprise offre un Puppet master completamente gestito, una suite di strumenti di automazione che ti consentono di ispezionare, distribuire, utilizzare e rendere le tue applicazioni a prova di futuro e l'accesso a un'interfaccia utente che ti consente di visualizzare informazioni sui tuoi nodi e sulle attività di Puppet. OpsWorks for Puppet Enterprise consente di utilizzare Puppet per automatizzare il modo in cui i nodi vengono configurati, distribuiti e gestiti, che si tratti di EC2 istanze Amazon o dispositivi locali. Un master OpsWorks for Puppet Enterprise offre un'automazione completa gestendo attività quali configurazioni di software e

sistemi operativi, installazioni di pacchetti, configurazioni di database, gestione delle modifiche, applicazione delle politiche, monitoraggio e garanzia della qualità.

OpsWorks Poiché Puppet Enterprise gestisce il software Puppet Enterprise, è possibile eseguire automaticamente il backup del server in un momento a scelta, esegue sempre la versione più recente di Puppet compatibile con AWS e applica sempre gli aggiornamenti di sicurezza più recenti. Puoi utilizzare i gruppi Amazon EC2 Auto Scaling per associare automaticamente nuovi EC2 nodi Amazon al tuo server.

AWS OpsWorks per Chef Automate

AWS OpsWorks for Chef Automate ti consente di creare server Chef AWS gestiti che includono le funzionalità premium di Chef Automate e di utilizzare Chef DK e altri strumenti Chef per gestirli. Un server Chef gestisce i nodi nel tuo ambiente, archivia informazioni su di essi e funge da repository centrale per i libri di ricette di Chef. Questi libri contengono ricette che vengono eseguite dall'agente client Chef Infra (`chef-client`) su ogni nodo gestito tramite Chef. Puoi utilizzare strumenti Chef come Test Kitchen per gestire nodi `knife`e libri di cucina su un server Chef incluso nel servizio. AWS OpsWorks for Chef Automate

Chef Automate è un pacchetto software server incluso che fornisce un flusso di lavoro automatizzato per la distribuzione continua e i controlli di conformità. AWS OpsWorks for Chef Automate installa e gestisce Chef Automate, Chef Infra e Chef InSpec utilizzando una singola istanza Amazon Elastic Compute Cloud. Con AWS OpsWorks for Chef Automate, puoi usare libri di cucina Chef creati dalla community o personalizzati senza apportare modifiche specifiche. AWS OpsWorks

Poiché AWS OpsWorks for Chef Automate gestisce i componenti di Chef Automate su una singola istanza, è possibile eseguire automaticamente il backup del server in un momento a scelta, esegue sempre la versione secondaria più recente di Chef e applica sempre gli aggiornamenti di sicurezza più recenti. Puoi utilizzare i gruppi Amazon EC2 Auto Scaling per associare automaticamente nuovi EC2 nodi Amazon al tuo server.

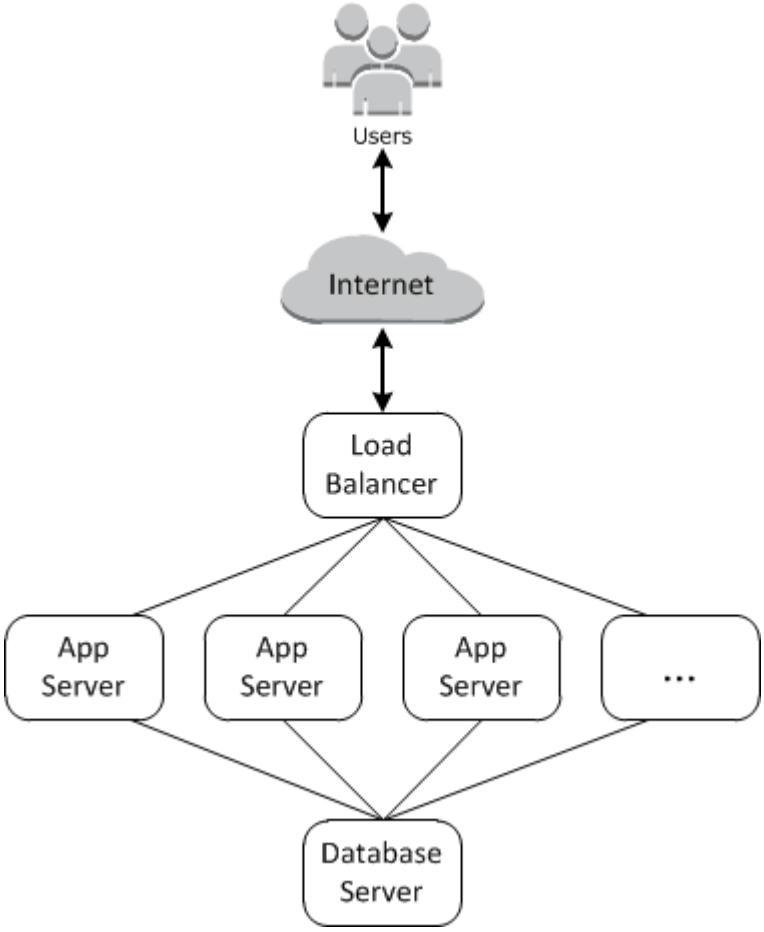
AWS OpsWorks Pile

L'elaborazione basata sul cloud di solito coinvolge gruppi di risorse AWS, come EC2 istanze e istanze Amazon Relational Database Service (RDS). Ad esempio, un'applicazione Web richiede in genere server applicazioni, server di database, sistemi di bilanciamento del carico e altre risorse. Questo gruppo di istanze viene chiamato in genere stack.

AWS OpsWorks Stacks, il servizio originale, offre un modo semplice e flessibile per creare e gestire stack e applicazioni. AWS OpsWorks Stacks ti consente di distribuire e monitorare le

applicazioni nei tuoi stack. Puoi creare stack che consentono di gestire le risorse cloud in gruppi specializzati denominati livelli. Un livello rappresenta un insieme di EC2 istanze che servono a uno scopo particolare, come servire applicazioni o ospitare un server di database. I livelli dipendono dalle ricette Chef per gestire attività quali l'installazione di pacchetti nelle istanze, la distribuzione di app e l'esecuzione di script.

Al contrario AWS OpsWorks for Chef Automate, AWS OpsWorks Stacks non richiede né crea server Chef; AWS OpsWorks Stacks esegue parte del lavoro di un server Chef per te. AWS OpsWorks Stacks monitora lo stato delle istanze e fornisce nuove istanze, se necessario, utilizzando Auto Healing e Auto Scaling. Uno stack di un server applicazioni semplice può essere analogo al diagramma seguente.



AWS OpsWorks per Puppet Enterprise

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

OpsWorks for Puppet Enterprise ti consente di avviare un master Puppet Enterprise in pochi minuti e di AWS OpsWorks gestirne le operazioni, i backup, i ripristini e gli aggiornamenti software. OpsWorks for Puppet Enterprise ti consente di concentrarti sulle attività principali di gestione della configurazione, anziché gestire un Puppet master. Utilizzando OpsWorks for Puppet Enterprise, puoi utilizzare le stesse configurazioni per gestire sia l'infrastruttura locale che quella cloud, aiutandoti a scalare in modo efficiente le tue operazioni in un ambiente ibrido. La gestione del server master Puppet viene semplificata dalla console Puppet Enterprise, la AWS Management Console e la AWS CLI.

Un master Puppet gestisce la configurazione dei nodi nell'ambiente tramite l'elaborazione di cataloghi di configurazione per nodi specifici per il software `puppet-agent` e funge da archivio centrale per i moduli di Puppet. Un Puppet master in OpsWorks per Puppet Enterprise viene distribuito nei nodi gestiti e fornisce `puppet-agent` le funzionalità premium di Puppet Enterprise.

Un master OpsWorks for Puppet Enterprise viene eseguito su un'istanza Amazon Elastic Compute Cloud. OpsWorks i server for Puppet Enterprise sono configurati per eseguire la versione più recente di Amazon Linux (Amazon Linux 2) e la versione più recente di Puppet Enterprise Master, versione 2019.8.5. Per ulteriori informazioni sulle modifiche apportate a Puppet Enterprise 2019.8.5, consulta le note di rilascio di Puppet Enterprise.

Quando le nuove versioni del software Puppet diventano disponibili, la manutenzione del sistema viene progettata per aggiornare automaticamente la versione di Puppet Enterprise sul server, non appena supera il test AWS. AWS esegue test approfonditi per verificare che gli aggiornamenti di Puppet siano pronti per la produzione e non interrompano gli ambienti esistenti dei clienti.

Puoi connettere qualsiasi computer o EC2 istanza locale che esegue un sistema operativo supportato e dispone di accesso alla rete a un master for Puppet Enterprise. OpsWorks Il software agente `puppet` viene installato dal master Puppet sui nodi che si desidera gestire.

Argomenti

- Supporto regionale OpsWorks per Puppet Enterprise
- AWS OpsWorks for Puppet Enterprise Fine del ciclo di vita FAQs
- Guida introduttiva a OpsWorks for Puppet Enterprise
- Crea un AWS OpsWorks for Puppet Enterprise master utilizzando AWS CloudFormation
- Aggiorna un server OpsWorks for Puppet Enterprise per utilizzare un dominio personalizzato
- Lavorare con i tag sulle AWS OpsWorks for Puppet Enterprise risorse
- Esegui il backup e il ripristino di un OpsWorks server Puppet Enterprise
- Manutenzione del sistema OpsWorks per Puppet Enterprise
- Aggiungere nodi automaticamente in OpsWorks Puppet Enterprise
- Dissocia un nodo da un server OpsWorks for Puppet Enterprise
- Eliminare un file OpsWorks per Puppet Enterprise Server
- Come migrare un server OpsWorks for Puppet Enterprise su Amazon Elastic Compute Cloud (Amazon) EC2
- Registrazione OpsWorks per le chiamate API Puppet Enterprise con AWS CloudTrail
- Risoluzione dei problemi OpsWorks per Puppet Enterprise

Supporto regionale OpsWorks per Puppet Enterprise

I seguenti endpoint regionali supportano OpsWorks i master di Puppet Enterprise. OpsWorks for Puppet Enterprise crea risorse associate ai tuoi Puppet master, come profili di istanza, utenti e ruoli di servizio, nello stesso endpoint regionale del tuo Puppet master. Il Puppet master deve essere in un VPC. È possibile utilizzare un VPC che si crea o di cui si dispone già, oppure utilizzare il VPC predefinito.

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Regione Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)

- Regione Asia Pacifico (Tokyo)
- Regione Asia Pacifico (Singapore)
- Regione Asia Pacifico (Sydney)
- Regione Europa (Francoforte)
- Regione Europa (Irlanda)

AWS OpsWorks for Puppet Enterprise Fine del ciclo di vita FAQs

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Argomenti

- In che modo i clienti esistenti saranno influenzati da questa fine del ciclo di vita?
- Cosa succede ai miei server se non eseguo alcuna azione?
- AWS OpsWorks for Puppet Enterprise Accetta nuovi clienti?
- La fine del ciclo di vita influirà Regioni AWS su tutti allo stesso tempo?
- Per quale livello di supporto tecnico è disponibile AWS OpsWorks for Puppet Enterprise?
- Sono un cliente attuale di OpsWorks for Puppet Enterprise e ho bisogno di avviare un server in un account che in precedenza non utilizzava il servizio. Sono in grado di farlo?
- Saranno disponibili nuove funzionalità per? AWS OpsWorks for Puppet Enterprise

In che modo i clienti esistenti saranno influenzati da questa fine del ciclo di vita?

I clienti esistenti non saranno interessati fino al 31 marzo 2024, data di fine del ciclo di vita di Puppet OpsWorks Enterprise. Dopo la data di fine del ciclo di vita, i clienti non saranno più in grado di gestire i propri server utilizzando la OpsWorks console o l'API.

Cosa succede ai miei server se non eseguo alcuna azione?

A partire dal 31 marzo 2024, non sarai più in grado di gestire i tuoi server utilizzando la OpsWorks console o l'API. A quel punto, smetteremo di eseguire qualsiasi funzione di gestione continua per i tuoi server, come i backup o la manutenzione. Per limitare l'impatto sui clienti, lasceremo che le EC2 istanze in esecuzione eseguano il backup dei server Puppet Enterprise, ma le relative licenze non saranno più valide in quanto l'utilizzo non è più coperto (o fatturato) dal contratto di servizio for Puppet Enterprise. OpsWorks Se desideri continuare a gestire la tua infrastruttura con Puppet Enterprise, consulta Come migrare un server OpsWorks for Puppet Enterprise su Amazon Elastic Compute Cloud (Amazon). EC2

AWS OpsWorks for Puppet Enterprise Accetta nuovi clienti?

No. AWS OpsWorks for Puppet Enterprise non accetta più nuovi clienti.

La fine del ciclo di vita influirà Regioni AWS su tutti allo stesso tempo?

Sì. L'API e la console raggiungeranno la fine del loro ciclo di vita e saranno inutilizzabili a partire dal 31 marzo 2024 in tutte le regioni. Per un elenco delle aree Regioni AWS in cui AWS OpsWorks for Puppet Enterprise è disponibile, consulta l'Elenco dei servizi AWS regionali.

Per quale livello di supporto tecnico è disponibile AWS OpsWorks for Puppet Enterprise?

AWS continuerà a fornire lo stesso livello di supporto AWS OpsWorks for Puppet Enterprise che i clienti hanno oggi fino alla data di fine del ciclo di vita. Se hai domande o dubbi, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Sono un cliente attuale di OpsWorks for Puppet Enterprise e ho bisogno di avviare un server in un account che in precedenza non utilizzava il servizio. Sono in grado di farlo?

Generalmente no, a meno che non vi siano circostanze eccezionali per farlo. Se hai una situazione particolare, contatta il Supporto AWS Team su AWS Re:post o tramite Premium AWS Support con i dettagli e la giustificazione e esamineremo la tua richiesta.

Saranno disponibili nuove funzionalità per? AWS OpsWorks for Puppet Enterprise

No. Poiché il servizio sta raggiungendo la fine del ciclo di vita, non rilasceremo alcuna nuova funzionalità. Tuttavia, continueremo a migliorare la sicurezza e a gestire i server come previsto fino alla data di fine del ciclo di vita.

Guida introduttiva a OpsWorks for Puppet Enterprise

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

OpsWorks for Puppet Enterprise ti consente di eseguire un server Puppet Enterprise in. AWSÈ possibile effettuare il provisioning di un server master Puppet Enterprise in circa 15 minuti.

A partire dal 3 maggio 2021, OpsWorks for Puppet Enterprise memorizza alcuni attributi del server Puppet Enterprise in. AWS Secrets Manager Per ulteriori informazioni, consulta Integrazione con AWS Secrets Manager.

La procedura dettagliata seguente ti aiuta a creare il tuo primo Puppet master per Puppet Enterprise. OpsWorks

Prerequisiti

Prima di iniziare, devi completare i seguenti prerequisiti.

Argomenti

- Installare il kit di sviluppo Puppet
- Installare gli strumenti client di Puppet Enterprise
- Impostazione di un archivio di controllo Git
- Impostare una VPC

- Configura una coppia di EC2 chiavi (opzionale)
- Prerequisiti per l'utilizzo di un dominio personalizzato (opzionale)

Installare il kit di sviluppo Puppet

1. Dal sito Web Puppet, scaricare il kit di sviluppo Puppet che corrisponde al sistema operativo del computer locale.
2. Installare il kit di sviluppo Puppet.
3. Aggiungere il kit di sviluppo Puppet alla variabile `PATH` del computer locale.

   - Su un sistema operativo Linux o macOS, è possibile aggiungere il kit di sviluppo Puppet alla variabile `PATH` eseguendo il seguente comando in una shell Bash.

     ```
     echo 'export PATH=/opt/puppetlabs/pdk/bin/pdk:$PATH' >> ~/.bash_profile && source
      ~/.bash_profile
     ```

   - In un sistema operativo basato su Windows, è possibile aggiungere il Puppet Development Kit alla **`PATH`** variabile utilizzando il seguente comando.NET Framework in una PowerShell sessione o nella finestra di dialogo Variabili di ambiente accessibile da Proprietà del sistema. Potrebbe essere necessario eseguire la PowerShell sessione come amministratore per eseguire il comando seguente.

     ```
     [Environment]::SetEnvironmentVariable("Path","new path value","Machine")
     ```

Installare gli strumenti client di Puppet Enterprise

Gli strumenti client Puppet Enterprise (PE) sono un set di strumenti della riga di comando che consentono di accedere ai servizi Puppet Enterprise dalla workstation. Gli strumenti possono essere installati su diversi sistemi operativi e possono anche essere installati sui nodi che si gestiscono utilizzando Puppet. Per informazioni sui sistemi operativi supportati per gli strumenti e su come installarli, vedere Installazione degli strumenti client PE nella documentazione Puppet Enterprise.

Impostazione di un archivio di controllo Git

Prima di poter avviare un Puppet master, è necessario disporre di un archivio di controllo configurato in Git per memorizzare e gestire i moduli e le classi Puppet. Per la procedura di avvio del server master Puppet Enterprise sono necessari l'URL di un repository Git e informazioni relative all'account

HTTPS o SSH per accedere al repository. Per ulteriori informazioni su come configurare un archivio di controllo che il Puppet Enterprise master utilizzerà, consulta Impostazione di un archivio di controllo. Puoi anche trovare le istruzioni per la configurazione del repository di controllo nel readme del repository di `control-repo`esempio di Puppet su. GitHub La struttura dell'archivio di controllo è simile a quanto segue.

```
### LICENSE
### Puppetfile
### README.md
### environment.conf
### hieradata
#   ### common.yaml
#   ### nodes
#       ### example-node.yaml
### manifests
#   ### site.pp
### scripts
#   ### code_manager_config_version.rb
#   ### config_version.rb
#   ### config_version.sh
### site
    ### profile
    #   ### manifests
    #       ### base.pp
    #       ### example.pp
    ### role
        ### manifests
            ### database_server.pp
            ### example.pp
            ### webserver.pp
```

Configurare un repository utilizzando CodeCommit

È possibile creare un nuovo repository utilizzando. CodeCommit Per ulteriori informazioni su come CodeCommit creare il repository di controllo, consulta questa the section called “Facoltativo: Usa CodeCommit” guida. Per ulteriori informazioni su come iniziare a usare Git on CodeCommit, consulta Getting started with AWS CodeCommit. Per autorizzare il server OpsWorks for Puppet Enterprise per il tuo repository, collega la `AWSCodeCommitReadOnly` policy al ruolo del profilo dell'istanza IAM.

Impostare una VPC

Il tuo master OpsWorks for Puppet Enterprise deve operare in un Amazon Virtual Private Cloud. Puoi aggiungerlo in una VPC, utilizzare la VPC predefinita, oppure creare una nuova VPC che contenga il server. Per informazioni su Amazon VPC e su come creare un nuovo VPC, consulta la Amazon VPC Getting Started Guide.

Se utilizzi una VPC esistente o ne crei una nuova, questa deve disporre delle seguenti impostazioni o proprietà.

- Il VPC deve avere almeno una sottorete.

  Se il tuo master OpsWorks for Puppet Enterprise sarà accessibile al pubblico, rendi pubblica la sottorete e abilita l'assegnazione automatica dell'IP pubblico.
- La DNS resolution (Risoluzione DNS) deve essere abilitata.
- Nella sottorete, abilitare Auto-assign Public IP (Assegna automaticamente IP pubblico).

Configura una coppia di EC2 chiavi (opzionale)

Una connessione SSH non è necessaria o consigliata per la gestione tipica del server Puppet; puoi utilizzare i AWS CLI comandi AWS Management Console and per eseguire molte attività di gestione sul tuo server Puppet.

È necessaria una coppia di EC2 chiavi per connettersi al server tramite SSH nel caso in cui si perda o si desideri modificare la password di accesso per la console basata sul web di Puppet Enterprise. Puoi utilizzare una coppia di chiavi esistente o crearne una nuova. Per ulteriori informazioni su come creare una nuova coppia di EC2 chiavi, consulta Amazon EC2 Key Pairs.

Se non hai bisogno di una EC2 key pair, sei pronto per creare un master Puppet Enterprise.

Prerequisiti per l'utilizzo di un dominio personalizzato (opzionale)

È possibile configurare il master Puppet Enterprise nel proprio dominio, specificando un endpoint pubblico in un dominio personalizzato da utilizzare come endpoint del server. Quando si utilizza un dominio personalizzato, sono necessarie tutte le seguenti operazioni, come descritto nel dettaglio in questa sezione.

Argomenti

- Configurare un dominio personalizzato
- Ottenere un certificato.
- Ottenere una chiave privata

Configurare un dominio personalizzato

Per eseguire il server Chef Automate sul proprio dominio personalizzato, è necessario un endpoint pubblico di un server, ad esempio `https://aws.my-company.com`. Se si specifica un dominio personalizzato, è necessario fornire anche un certificato e una chiave privata, come descritto nelle sezioni precedenti.

Per accedere al server dopo la creazione, aggiungere un record DNS CNAME nel servizio DNS preferito. Questo record deve puntare il dominio personalizzato all'endpoint (il valore dell'attributo `Endpoint` del server) generato dal processo di creazione del master Puppet. Non è possibile accedere al server utilizzando il valore `Endpoint` generato se il server utilizza un dominio personalizzato.

Ottenere un certificato.

Per configurare il master Puppet sul proprio dominio personalizzato, è necessario un certificato HTTPS in formato PEM. Può essere un singolo certificato autofirmato o una catena di certificati. Quando si completa il flusso di lavoro Crea un master Puppet Enterprise, se si specifica questo certificato, è necessario fornire anche un dominio e una chiave privata personalizzati.

Di seguito sono riportati i requisiti per il valore del certificato:

- È possibile fornire un certificato personalizzato autofirmato o l'intera catena di certificati.
- Il certificato deve essere un certificato X509 valido o una catena di certificati in formato PEM.
- Il certificato deve essere valido al momento del caricamento. Non è possibile utilizzare un certificato prima dell'inizio del periodo di validità (la data `NotBefore` del certificato) o dopo la data di scadenza (la data `NotAfter` del certificato).
- Il nome comune o i nomi alternativi del soggetto del certificato (SANs), se presenti, devono corrispondere al valore del dominio personalizzato.
- Il certificato deve corrispondere al valore del campo Custom private key (Chiave privata personalizzata).

Ottenere una chiave privata

Per configurare il master Puppet nel proprio dominio personalizzato, è necessaria una chiave privata in formato PEM per la connessione al server tramite HTTPS. La chiave privata non deve essere crittografata; non può essere protetta da password o da passphrase. Se si specifica una chiave privata personalizzata, è necessario fornire anche un dominio e un certificato personalizzati.

Creare un Puppet Enterprise Master

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

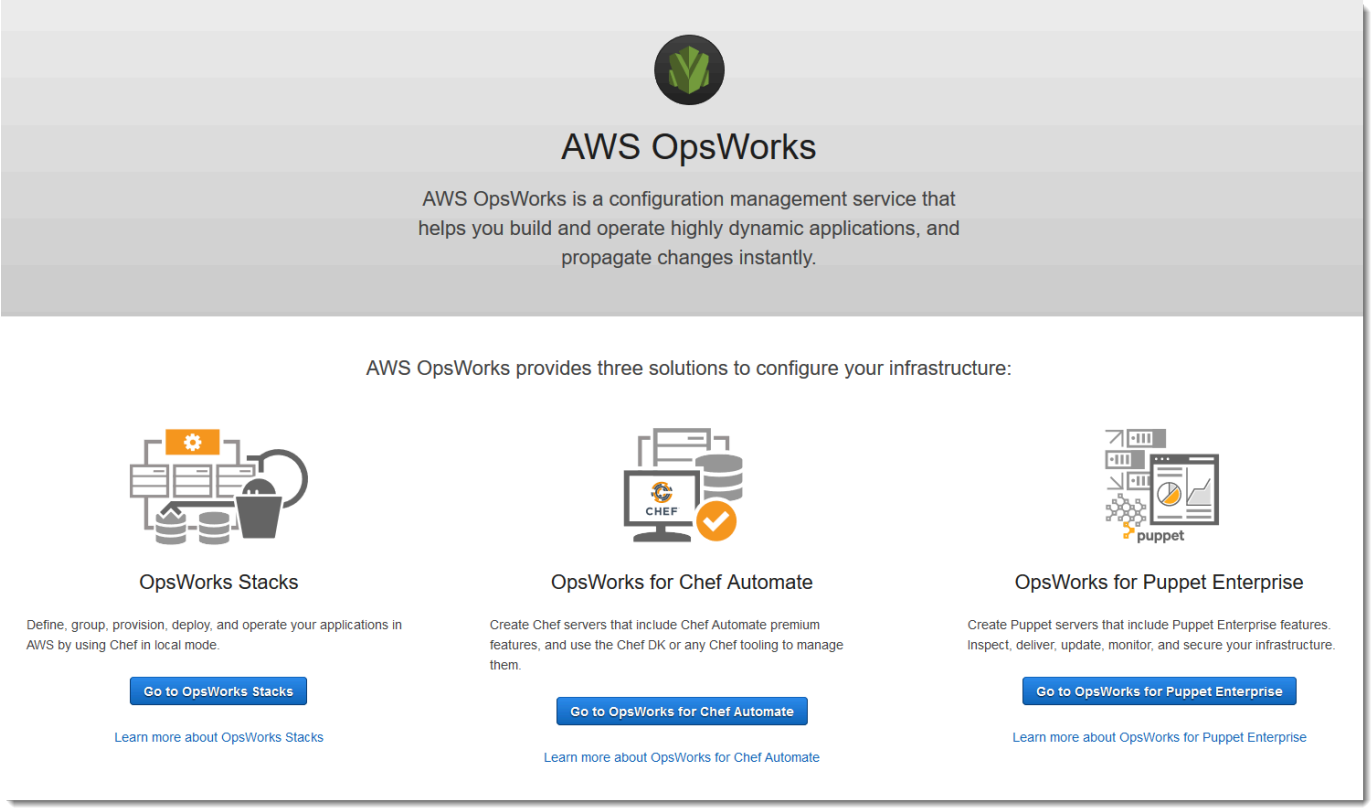
Puoi creare un Puppet master utilizzando la console OpsWorks for Puppet Enterprise o il. AWS CLI

Argomenti

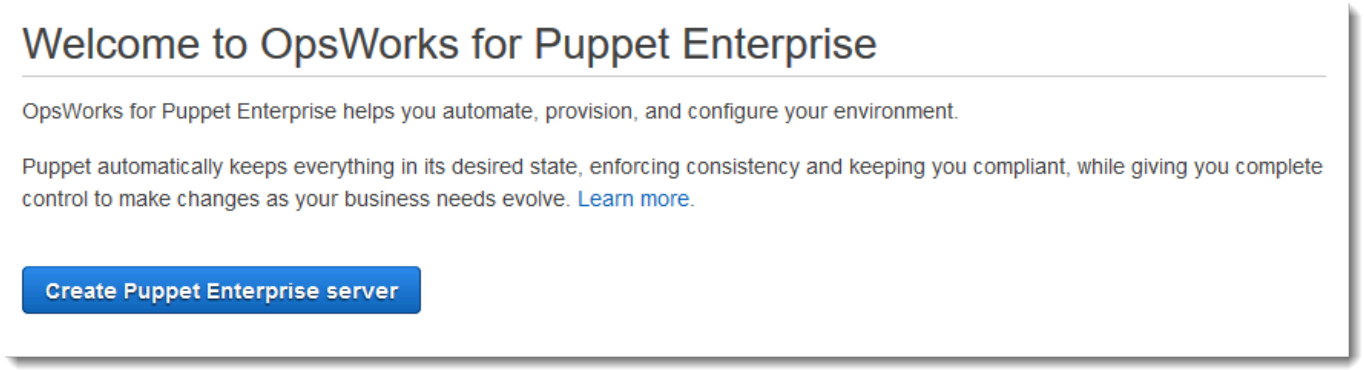
- Crea un Puppet Enterprise Master utilizzando il AWS Management Console
- Crea un Puppet Enterprise Master utilizzando il AWS CLI

Crea un Puppet Enterprise Master utilizzando il AWS Management Console

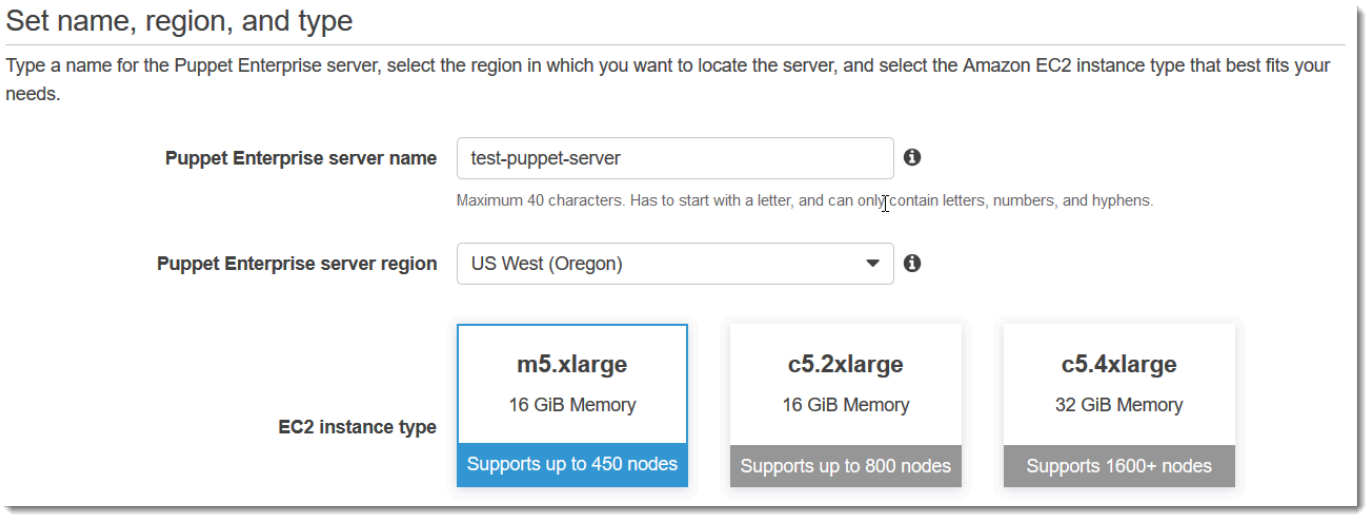
1. Accedi a AWS Management Console e apri la AWS OpsWorks console all'indirizzo https://console.aws.amazon.com/opsworks/.
2. Nella AWS OpsWorks home page, scegli Vai a OpsWorks per Puppet Enterprise.



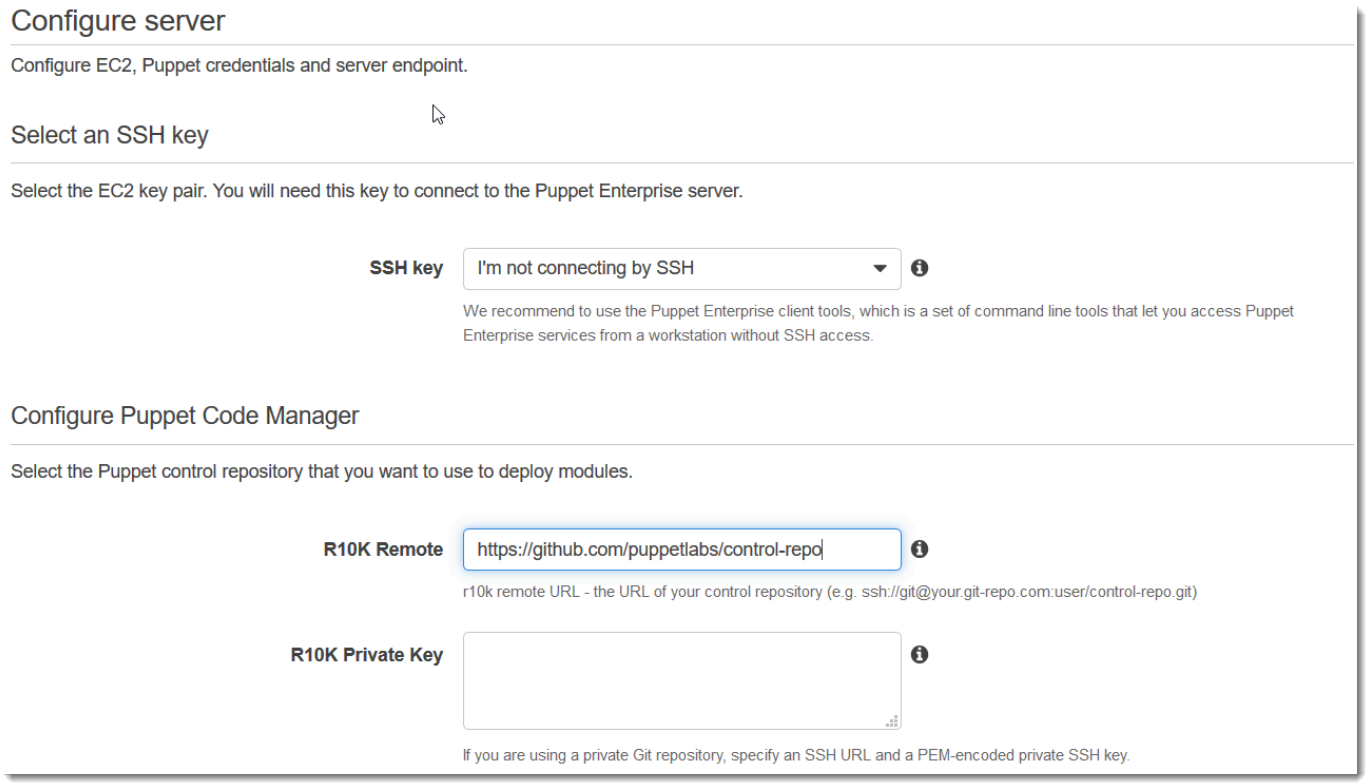
3. Nella home page OpsWorks di Puppet Enterprise, scegli Crea server Puppet Enterprise.



4. Nella pagina Set name, region, and type (Imposta nome, regione e tipo) specificare un nome per il server. I nomi dei Puppet master possono avere un massimo di 40 caratteri, devono iniziare con una lettera e possono contenere solo caratteri alfanumerici e trattini. Seleziona una regione supportata, quindi scegli un tipo di istanza che supporti il numero di nodi che desideri gestire. Se necessario, puoi modificare il tipo di istanza dopo la creazione del tuo server. Per questa procedura dettagliata, stiamo creando un tipo di istanza m5.xlarge nella regione Stati Uniti occidentali (Oregon). Scegli Next (Successivo).



5. Nella pagina Configure server (Configura server), lasciare la selezione predefinita nell'elenco a discesa SSH key (Chiave SSH), a meno che non si desideri specificare il nome di una coppia di chiavi. Nel campo r10k remote (r10k remoto) dell'area Configure Puppet Code Manager (Configura Puppet Code Manager), specificare un URL SSH o HTTPS valido del Git in remoto. Nel campo della chiave privata r10k, incolla la chiave privata SSH che AWS OpsWorks puoi utilizzare per accedere all'archivio remoto r10k. Questa viene fornita da Git al momento della creazione di un repository privato, ma non è necessaria se si sta utilizzando l'autenticazione HTTPS per accedere al repository di controllo. Scegli Next (Successivo).

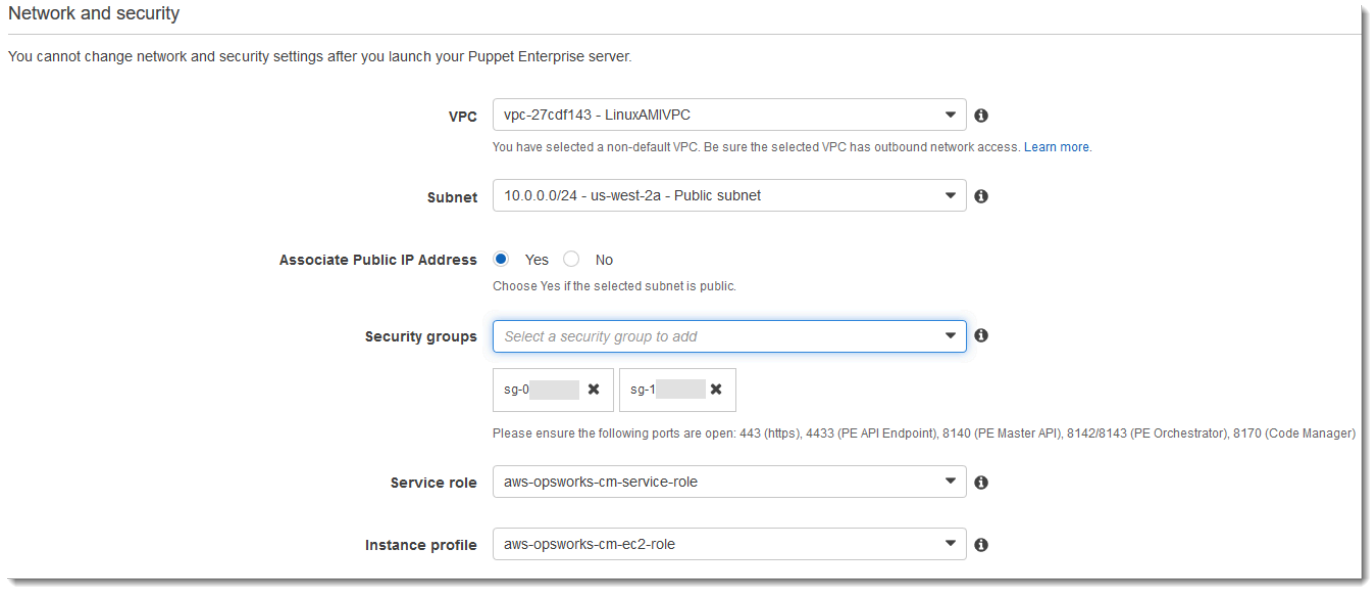


6. Per Specify server endpoint (Specifica endpoint del server), lasciare l'impostazione predefinita, Use an automatically-generated endpoint (Utilizza un endpoint generato automaticamente) e quindi scegliere Next (Avanti), a meno che non si desideri che il server sia in un dominio personalizzato di proprietà dell'utente. Per configurare un dominio personalizzato, passare alla fase successiva.
7. Per utilizzare un dominio personalizzato, per Specify server endpoint (Specifica endpoint del server), scegliere Use a custom domain (Usa un dominio personalizzato) dall'elenco a discesa.

    a. Per Fully qualified domain name (FQDN) (Nome di dominio completo (FQDN)), specificare un nome completo (FQDN). È necessario essere proprietari del nome di dominio che si desidera utilizzare.

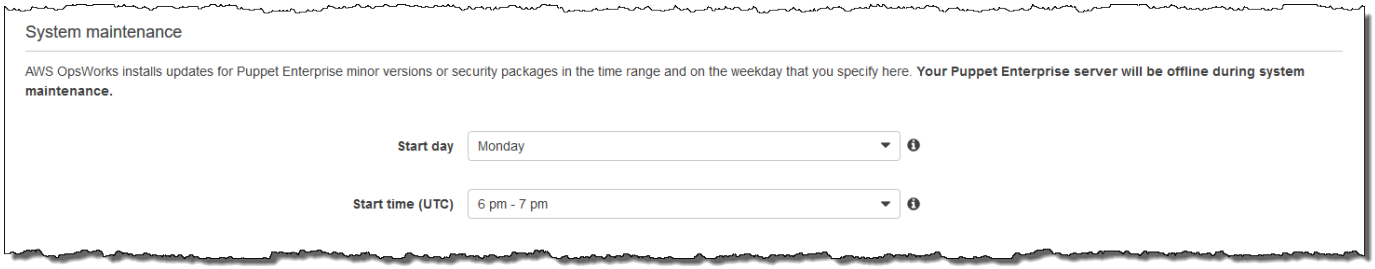
    b. Per SSL certificate (Certificato SSL), incollare l'intero certificato in formato PEM, iniziando con `-----BEGIN CERTIFICATE-----` e terminando con `-----END CERTIFICATE-----`. L'oggetto del certificato SSL deve corrispondere al nome di dominio completo immesso nella fase precedente. Rimuovere eventuali righe aggiuntive prima e dopo il certificato.

    c. Per SSL private key (Chiave privata SSL), incollare l'intera chiave privata RSA, iniziando con `-----BEGIN RSA PRIVATE KEY-----` e finendo con `-----END RSA PRIVATE KEY-----`. La chiave privata SSL deve corrispondere alla chiave pubblica nel certificato SSL immesso nella fase precedente. Rimuovi eventuali linee extra prima e dopo la chiave privata. Scegli Next (Successivo).
8. Nella pagina Configura impostazioni avanzate, nell'area Rete e sicurezza, scegli un VPC, una sottorete e uno o più gruppi di sicurezza. AWS OpsWorks può generare automaticamente un gruppo di sicurezza, un ruolo di servizio e un profilo di istanza, se non ne hai già uno che desideri utilizzare. Il tuo server può essere un membro di più gruppi di sicurezza. Non è possibile modificare le impostazioni di rete e di sicurezza per il Puppet master dopo aver lasciato questa pagina.

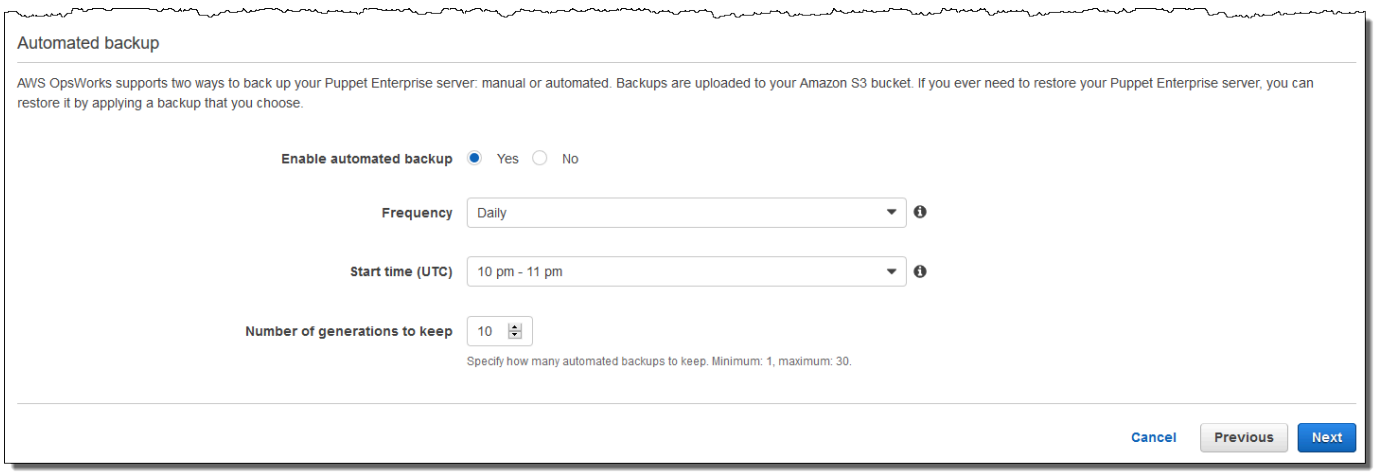


9. Nella sezione System maintenance (Manutenzione sistema), impostare il giorno e l'ora quando si desidera che il mantenimento del sistema inizi. In quanto ci si deve aspettare che il server sia offline durante il mantenimento del sistema, scegliere un orario di utilizzo del server basso durante l'orario di ufficio regolare.

   La finestra di manutenzione è obbligatoria. È possibile modificare il giorno e l'ora di inizio in un secondo momento utilizzando AWS Management Console AWS CLI, o APIs.



10. Configurare i backup. I backup automatici sono abilitati per impostazione predefinita. Imposta una frequenza e un'ora preferite per l'avvio del backup automatico e imposta il numero di generazioni di backup da archiviare in Amazon Simple Storage Service. È possibile conservare un massimo di 30 backup; quando viene raggiunto il numero massimo, OpsWorks for Puppet Enterprise elimina i backup più vecchi per fare spazio a quelli nuovi.



11. (Facoltativo) In Tag, aggiungi tag al server e alle risorse correlate, come l' EC2 istanza, l'indirizzo IP elastico, il gruppo di sicurezza, il bucket S3 e i backup. Per ulteriori informazioni sull'etichettatura di un server OpsWorks for Puppet Enterprise, consulta. Lavorare con i tag sulle AWS OpsWorks for Puppet Enterprise risorse
12. Quando la configurazione delle impostazioni avanzate è terminata, selezionare Next (Successivo).
13. Nella pagina Review (Esamina), esaminare le proprie opzioni. Quando si è pronti per creare il server, selezionare Launch (Avvia).

    Mentre aspetti di creare il tuo Puppet master, vai AWS OpsWorks a scaricare lo Starter Kit Configurare il Puppet Master utilizzando lo Starter Kit e le credenziali della console Puppet Enterprise. Non attendere che il server sia online per scaricare questi elementi.

    Al termine della creazione del server, il tuo Puppet master è disponibile nella OpsWorks home page di Puppet Enterprise, con lo stato online. Dopo che il server è online, la console Puppet Enterprise è disponibile nel dominio del server, in un URL nel formato seguente: `https://`*`your_server_name`*`-`*`randomID`*`.`*`region`*`.opsworks-cm.io`.

Crea un Puppet Enterprise Master utilizzando il AWS CLI

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile.

Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

La creazione di un server master OpsWorks for Puppet Enterprise eseguendo AWS CLI comandi è diversa dalla creazione di un server nella console. Nella console, AWS OpsWorks crea automaticamente un ruolo di servizio e un gruppo di sicurezza, se non specifichi quelli esistenti che desideri utilizzare. In AWS CLI, AWS OpsWorks può creare automaticamente un gruppo di sicurezza se non lo si specifica, ma non crea automaticamente un ruolo di servizio; è necessario fornire un ruolo di servizio ARN come parte del `create-server` comando. Nella console, mentre AWS OpsWorks crei il tuo Puppet master, scarichi lo starter kit e le credenziali di accesso per la console Puppet Enterprise. Poiché non è possibile farlo quando si crea un master OpsWorks for Puppet Enterprise utilizzando AWS CLI, si utilizza un'utilità di elaborazione JSON per ottenere le credenziali di accesso e lo starter kit dai risultati del `create-server` comando dopo che il master nuovo per Puppet Enterprise è online. OpsWorks

Se sul tuo computer locale non è già in esecuzione AWS CLI, scaricalo e installalo AWS CLI seguendo le istruzioni di installazione nella AWS Command Line Interface User Guide. Questa sezione non descrive tutti i parametri che è puoi utilizzare con il comando `create-server`. Per ulteriori informazioni sui parametri `create-server`, consulta `create-server` nella documentazione di riferimento delle AWS CLI .

1. Assicurarsi di completare Prerequisiti. Per creare il Puppet master, è necessario un ID di sottorete, quindi è necessario disporre di un VPC.
2. Crea un ruolo di servizio e un profilo di istanza.
3. Trova e copia i ruoli ARNs di servizio nel tuo account.

   ```
   aws iam list-roles --path-prefix "/service-role/" --no-paginate
   ```

   Nei risultati del comando `list-roles`, cercare le voci ARN dei ruoli di servizio che sono simili a quanto segue. Prendi nota del ruolo di servizio ARNs. Per creare il master Puppet Enterprise, hai bisogno di questi valori.

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

```
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "ec2.amazonaws.com"
                }
            }
        ]
    },
    "RoleId": "AROZZZZZZZZZQG6R22HC",
    "CreateDate": "2018-01-05T20:42:20Z",
    "RoleName": "aws-opsworks-cm-ec2-role",
    "Path": "/service-role/",
    "Arn": "arn:aws:iam::000000000000:role/service-role/aws-opsworks-cm-ec2-role"
},
{
    "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "opsworks-cm.amazonaws.com"
                }
            }
        ]
    },
    "RoleId": "AROZZZZZZZZZZZZZZZ6QE",
    "CreateDate": "2018-01-05T20:42:20Z",
    "RoleName": "aws-opsworks-cm-service-role",
    "Path": "/service-role/",
    "Arn": "arn:aws:iam::000000000000:role/service-role/aws-opsworks-cm-service-
role"
}
```

4. Trova e copia i profili ARNs delle istanze nel tuo account.

```
aws iam list-instance-profiles --no-paginate
```

Nei risultati del comando `list-instance-profiles`, cerca le voci ARN dei profili delle istanze che sono simili a quanto segue. Prendi nota del profilo dell'istanza ARNs. Per creare il master Puppet Enterprise, hai bisogno di questi valori.

```
{
    "Path": "/",
    "InstanceProfileName": "aws-opsworks-cm-ec2-role",
    "InstanceProfileId": "EXAMPLEDC6UR3LTUW7VHK",
    "Arn": "arn:aws:iam::123456789012:instance-profile/aws-opsworks-cm-ec2-role",
    "CreateDate": "2017-01-05T20:42:20Z",
    "Roles": [
        {
            "Path": "/service-role/",
            "RoleName": "aws-opsworks-cm-ec2-role",
            "RoleId": "EXAMPLEE4STNUQG6R22HC",
            "Arn": "arn:aws:iam::123456789012:role/service-role/aws-opsworks-cm-
ec2-role",
            "CreateDate": "2017-01-05T20:42:20Z",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            }
        }
    ]
},
```

5. Crea il master OpsWorks for Puppet Enterprise eseguendo il `create-server` comando.

   - Il `--engine` valore è `PuppetMonolithic`, `--engine-model` è e `--engine-version` può essere `2019` o`2017`.
   - Il nome del server deve essere univoco all'interno del tuo AWS account, all'interno di ogni regione. I nomi dei server devono iniziare con una lettera; quindi sono consentiti lettere, numeri o trattini (-) fino a un massimo di 40 caratteri.
   - Utilizza l'ARN del profilo dell'istanza e l'ARN del ruolo di servizio copiati nella Fase 3 e 4.

- I tipi di istanza validi sono `m5.xlarge`, `c5.2xlarge` o `c5.4xlarge`. Per ulteriori informazioni sulle specifiche di questi tipi di istanze, consulta Instance Types nella Amazon EC2 User Guide.
- Il parametro `--engine-attributes` è opzionale; se non si specifica una password di amministratore Puppet, il processo di creazione del server ne genera uno. Se si aggiunge `--engine-attributes`, specificare una `PUPPET_ADMIN_PASSWORD`, una password di amministratore per l'accesso alla pagina Web della console Puppet Enterprise. La password deve utilizzare da 8 a 32 caratteri ASCII.
- Una coppia di chiavi SSH è facoltativa, ma può aiutare a connettersi al Puppet master se si deve reimpostare la password di amministratore della console. Per ulteriori informazioni sulla creazione di una coppia di chiavi SSH, consulta Amazon EC2 Key Pairs nella Amazon EC2 User Guide.
- Per utilizzare un dominio personalizzato, aggiungere i seguenti parametri al comando. Diversamente, il processo di creazione del master Puppet genera automaticamente un endpoint. Tutti e tre i parametri sono necessari per configurare un dominio personalizzato. Per informazioni sui requisiti aggiuntivi per l'utilizzo di questi parametri, consulta CreateServeril AWS OpsWorks CM API Reference.
  - `--custom-domain` - Un endpoint pubblico opzionale di un server, ad esempio `https://aws.my-company.com`.
  - `--custom-certificate` - Un certificato HTTPS in formato PEM. Il valore può essere un singolo certificato autofirmato o una catena di certificati.
  - `--custom-private-key` - Una chiave privata in formato PEM per la connessione al server tramite HTTPS. La chiave privata non deve essere crittografata; non può essere protetta da password o da passphrase.
- La manutenzione del sistema settimanale è obbligatoria. I valori validi devono essere specificati nel seguente formato: `DDD:HH:MM`. L'ora specificata è espressa nel Tempo coordinato universale (UTC). Se non si specifica un valore per `--preferred-maintenance-window`, il valore predefinito è un periodo casuale di un'ora il martedì, mercoledì o venerdì.
- I valori validi per `--preferred-backup-window` devono essere specificati in uno dei seguenti formati: `HH:MM` per backup giornaliero o `DDD:HH:MM` per backup settimanali. L'orario specificato è nel formato UTC. Il valore predefinito è un'ora di avvio giornaliera casuale. Per disattivare i backup automatici, aggiungere invece il parametro `--disable-automated-backup`.

- Per `--security-group-ids`, inserisci uno o più gruppi di sicurezza IDs, separati da uno spazio.
- Per `--subnet-ids`, inserire un ID di sottorete.

```
aws opsworks-cm create-server --engine "Puppet" --engine-model "Monolithic"
 --engine-version "2019" --server-name "server_name" --instance-profile-arn
 "instance_profile_ARN" --instance-type "instance_type" --engine-attributes
 '{"PUPPET_ADMIN_PASSWORD":"ASCII_password"}' --key-pair "key_pair_name" --
preferred-maintenance-window "ddd:hh:mm" --preferred-backup-window "ddd:hh:mm"
 --security-group-ids security_group_id1 security_group_id2 --service-role-arn
 "service_role_ARN" --subnet-ids subnet_ID
```

Di seguito è riportato un esempio.

```
aws opsworks-cm create-server --engine "Puppet" --engine-model
 "Monolithic" --engine-version "2019" --server-name "puppet-02" --
instance-profile-arn "arn:aws:iam::111122223333:instance-profile/aws-
opsworks-cm-ec2-role" --instance-type "m5.xlarge" --engine-attributes
 '{"PUPPET_ADMIN_PASSWORD":"zZZzDj2DLYXSZFRv1d"}' --key-pair "amazon-test"
 --preferred-maintenance-window "Mon:08:00" --preferred-backup-window
 "Sun:02:00" --security-group-ids sg-b00000001 sg-b0000008 --service-role-arn
 "arn:aws:iam::111122223333:role/service-role/aws-opsworks-cm-service-role" --
subnet-ids subnet-383daa71
```

Nell'esempio seguente viene creato un master Puppet che utilizza un dominio personalizzato.

```
aws opsworks-cm create-server \
    --engine "Puppet" \
    --engine-model "Monolithic" \
    --engine-version "2019" \
    --server-name "puppet-02" \
    --instance-profile-arn "arn:aws:iam::111122223333:instance-profile/aws-
opsworks-cm-ec2-role" \
    --instance-type "m5.xlarge" \
    --engine-attributes '{"PUPPET_ADMIN_PASSWORD":"zZZzDj2DLYXSZFRv1d"}' \
    --custom-domain "my-puppet-master.my-corp.com" \
    --custom-certificate "-----BEGIN CERTIFICATE----- EXAMPLEqEXAMPLE== -----END
 CERTIFICATE-----" \
    --custom-private-key "-----BEGIN RSA PRIVATE KEY----- EXAMPLEqEXAMPLE= -----END
 RSA PRIVATE KEY-----" \
```

```
    --key-pair "amazon-test"
    --preferred-maintenance-window "Mon:08:00" \
    --preferred-backup-window "Sun:02:00" \
    --security-group-ids sg-b00000001 sg-b0000008 \
    --service-role-arn "arn:aws:iam::111122223333:role/service-role/aws-opsworks-
cm-service-role" \
    --subnet-ids subnet-383daa71
```

Nell'esempio seguente viene creato un master Puppet che aggiunge due tag: `Stage: Production` e `Department: Marketing`. Per ulteriori informazioni sull'aggiunta e la gestione dei tag sui OpsWorks server Puppet Enterprise, Lavorare con i tag sulle AWS OpsWorks for Puppet Enterprise risorse consulta questa guida.

```
aws opsworks-cm create-server \
    --engine "Puppet" \
    --engine-model "Monolithic" \
    --engine-version "2019" \
    --server-name "puppet-02" \
    --instance-profile-arn "arn:aws:iam::111122223333:instance-profile/aws-
opsworks-cm-ec2-role" \
    --instance-type "m5.xlarge" \
    --engine-attributes '{"PUPPET_ADMIN_PASSWORD":"zZZzDj2DLYXSZFRv1d"}' \
    --key-pair "amazon-test"
    --preferred-maintenance-window "Mon:08:00" \
    --preferred-backup-window "Sun:02:00" \
    --security-group-ids sg-b00000001 sg-b0000008 \
    --service-role-arn "arn:aws:iam::111122223333:role/service-role/aws-opsworks-
cm-service-role" \
    --subnet-ids subnet-383daa71 \
    --tags [{\"Key\":\"Stage\",\"Value\":\"Production\"},{\"Key\":\"Department\",
\"Value\":\"Marketing\"}]
```

6. OpsWorks for Puppet Enterprise impiega circa 15 minuti per creare un nuovo server. Non annullare l'output del comando `create-server` né chiudere la sessione shell, poiché l'output può contenere informazioni importanti che non vengono visualizzate nuovamente. Per ottenere le password e lo starter kit dai risultati `create-server`, passare alla fase successiva.

   Se si utilizza un dominio personalizzato con il server, nell'output del comando `create-server`, copiare il valore dell'attributo `Endpoint`. Di seguito è riportato un esempio.

```
"Endpoint": "puppet-07-exampleexample.opsworks-cm.us-east-1.amazonaws.com"
```

7. Se hai scelto di fare in modo che OpsWorks Puppet Enterprise generi una password per te, puoi estrarla in un formato utilizzabile dai `create-server` risultati utilizzando un processore JSON come jq. Dopo aver installato jq, è possibile eseguire i seguenti comandi per estrarre la password di amministratore Puppet e lo starter kit. Se non si fornisce la password nella Fase 3, assicurarsi di salvare la password di amministratore estratta in una posizione sicura, ma comoda.

```
#Get the Puppet password:
cat resp.json | jq -r '.Server.EngineAttributes[] | select(.Name ==
 "PUPPET_ADMIN_PASSWORD") | .Value'

#Get the Puppet Starter Kit:
cat resp.json | jq -r '.Server.EngineAttributes[] | select(.Name ==
 "PUPPET_STARTER_KIT") | .Value' | base64 -D > starterkit.zip
```

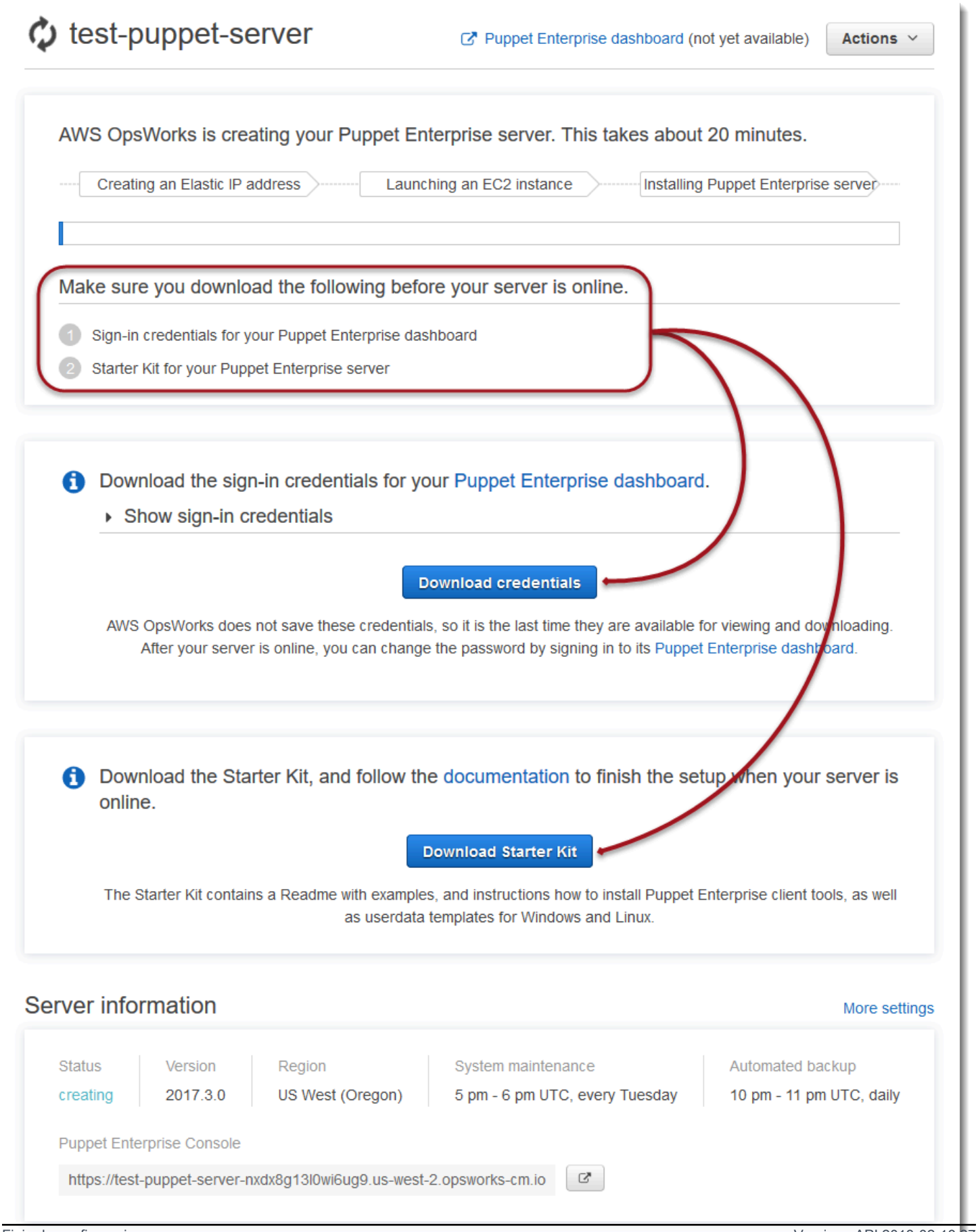
> **Note**
>
> Non è possibile rigenerare un nuovo starter kit Puppet master nella AWS Management Console. Quando crei un master Puppet utilizzando il AWS CLI, esegui il `jq` comando precedente per salvare lo starter kit con codifica base64 nei risultati come file ZIP. `create-server`

8. Se non si utilizza un dominio personalizzato, passare alla fase successiva. Se utilizzi un dominio personalizzato con il server, crea una voce CNAME nello strumento di gestione DNS dell'azienda per indirizzare il dominio personalizzato all'endpoint OpsWorks for Puppet Enterprise che hai copiato nel passaggio 6. Non è possibile raggiungere o eseguire l'accesso al server con un dominio personalizzato fino a quando non si completa questa fase.
9. Passare alla fase successiva, the section called “Finire la configurazione”.

Configurare il Puppet Master utilizzando lo Starter Kit

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Mentre la creazione del master Puppet è ancora in corso, la pagina delle proprietà del server si apre nella console OpsWorks for Puppet Enterprise. La prima volta che si lavora con un nuovo Puppet master, la pagina Proprietà chiede di scaricare due voci necessarie. Scaricare queste voci prima che il server Puppet sia online; i pulsanti di download non sono disponibili dopo che un nuovo server è online.


test-puppet-server
Puppet Enterprise dashboard (not yet available)
Actions
AWS OpsWorks is creating your Puppet Enterprise server. This takes about 20 minutes.
Creating an Elastic IP address
Launching an EC2 instance
Installing Puppet Enterprise server
Make sure you download the following before your server is online.
1 Sign-in credentials for your Puppet Enterprise dashboard
2 Starter Kit for your Puppet Enterprise server
Download the sign-in credentials for your Puppet Enterprise dashboard.
Show sign-in credentials
Download credentials
AWS OpsWorks does not save these credentials, so it is the last time they are available for viewing and downloading.
After your server is online, you can change the password by signing in to its Puppet Enterprise dashboard.
Download the Starter Kit, and follow the documentation to finish the setup when your server is online.
Download Starter Kit
The Starter Kit contains a Readme with examples, and instructions how to install Puppet Enterprise client tools, as well as userdata templates for Windows and Linux.
Server information
More settings
Status
creating
Version
2017.3.0
Region
US West (Oregon)
System maintenance
5 pm - 6 pm UTC, every Tuesday
Automated backup
10 pm - 11 pm UTC, daily
Puppet Enterprise Console
https://test-puppet-server-nxdx8g13l0wi6ug9.us-west-2.opsworks-cm.io

- Sign-in credentials for the Puppet master (Credenziali di accesso per il Puppet master). Utilizzerai queste credenziali per accedere alla console di Puppet Enterprise, dove esegui la maggior parte della gestione dei nodi. AWS OpsWorks non salva queste credenziali; questa è l'ultima volta che sono disponibili per la visualizzazione e il download. Se necessario, puoi modificare la password che viene fornita con queste credenziali dopo aver eseguito l'accesso.
- Starter Kit. Lo Starter Kit contiene un file README con informazioni ed esempi che descrivono come completare la configurazione e le credenziali di amministratore per la console Puppet Enterprise. Le nuove credenziali vengono generate e le vecchie credenziali invalidate, ogni volta che si scarica lo Starter Kit.

Prerequisiti

1. Mentre la creazione del server è ancora in corso, scaricare le credenziali di accesso per il Puppet master e salvarle in una posizione sicura ma comoda.
2. Scarica lo Starter Kit e decomprimi il suo file .zip nella directory del tuo WorkSpace. Non condividere le credenziali di accesso. Se altri utenti gestiranno il Puppet master, aggiungerli come amministratori nella console Puppet Enterprise più tardi. Per ulteriori informazioni su come aggiungere utenti al Puppet master, consulta Creazione e gestione di utenti e ruoli utente nella documentazione Puppet Enterprise.

Installare il certificato Puppet Master

Per lavorare con il Puppet master e aggiungere nodi da gestire, è necessario installare il certificato. Installarlo eseguendo il AWS CLI comando seguente. Non è possibile eseguire questa operazione in AWS Management Console.

```
aws --region region opsworks-cm describe-servers --server-name server_name --query
 "Servers[0].EngineAttributes[?Name=='PUPPET_API_CA_CERT'].Value" --output text
 > .config/ssl/cert/ca.pem
```

Generare un token a breve termine

Per utilizzare l'API Puppet, è necessario creare un token a breve termine per se stessi. Questa fase non è necessaria per utilizzare la console di Enterprise Puppet. Generare il token eseguendo il seguente comando:

La durata predefinita è di cinque minuti, ma è possibile modificare questa impostazione predefinita.

```
puppet-access login --config-file .config/puppetlabs/client-tools/puppet-access.conf --
lifetime 8h
```

> **Note**
>
> Poiché la durata predefinita è di cinque minuti, il comando di esempio precedente aggiunge il parametro `--lifetime` per estendere la durata del token per un periodo di tempo più lungo. È possibile impostare la durata del token per un periodo fino a 10 anni (`10y`). Per ulteriori informazioni su come modificare la durata predefinita del token, consulta Modifica della durata predefinita del token nella documentazione Puppet Enterprise.

Configurare l'esempio Apache dello Starter Kit

Dopo aver scaricato e decompresso lo Starter Kit, puoi utilizzare il ramo di esempio nella `control-repo-example` cartella di esempio inclusa per configurare un server web Apache sui tuoi nodi gestiti.

Lo Starter Kit include due cartelle `control-repo`: `control-repo` e `control-repo-example`. La `control-repo` cartella include un `production` ramo che è rimasto invariato rispetto a quello che vedresti nell'archivio Puppet. GitHub La `control-repo-example` cartella ha anche un `production` ramo che include un codice di esempio per configurare un server Apache con un sito Web di prova.

1. Spingere il ramo `control-repo-example` `production` al Git remoto (l'URL `r10k_remote` del tuo Puppet master). Nella directory principale dello Starter Kit, esegui quanto segue, sostituendolo *`r10kRemoteUrl`* con il tuo `r10k_remote` URL.

   ```
   cd  control-repo-example
   git remote add origin r10kRemoteUrl
   git push origin production
   ```

   Il Code Manager di Puppet utilizza i rami Git come ambienti. Come impostazione predefinita, tutti i nodi sono nell'ambiente di produzione.

   > **Important**
   >
   > Non spingere su un ramo `master`. Il ramo `master` è prenotato per il Puppet master.

2. Distribuire il codice nel rame `control-repo-example` nel Puppet master. Ciò permette al Puppet Master di scaricare il codice Puppet dall'archivio Git (`r10k_remote`). Nella directory root dello Starter Kit, eseguire quanto segue.

```
puppet-code deploy --all --wait --config-file .config/puppet-code.conf
```

Per ulteriori informazioni su come applicare la configurazione di esempio di Apache ai nodi gestiti che crei in Amazon EC2, consulta Fase 2. Creazione delle istanze tramite uno script di associazione automatica questa guida.

Aggiungere nodi che il Puppet Master può gestire

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Argomenti

- Eseguire chiamate API associateNode()
- Considerazioni per l'aggiunta di nodi locali
- Ulteriori informazioni

Il modo consigliato per aggiungere nodi è utilizzare l' AWS OpsWorks `associateNode()`API. Il master server Puppet Enterprise ospita un archivio che è possibile utilizzare per installare il software agente Puppet su nodi che si desidera gestire, sia che i nodi si trovino su computer fisici locali sia su macchine virtuali. Il software Puppet Agent per alcuni sistemi operativi viene installato sul server OpsWorks for Puppet Enterprise come parte del processo di avvio. La tabella seguente mostra gli agenti del sistema operativo disponibili sul server OpsWorks for Puppet Enterprise al momento del lancio.

Agenti del sistema operativo preinstallati

Sistema operativo supportato	Versioni
Ubuntu	16.04, 18.04, 20.04
Red Hat Enterprise Linux (RHEL)	6, 7, 8
Windows	Edizioni a 64 bit di tutte le versioni Windows supportate da Puppet

È possibile aggiungere `puppet-agent` al server per altri sistemi operativi. Si noti che la manutenzione del sistema eliminerà agenti aggiunti al server dopo l'avvio. Anche se la maggior parte dei nodi collegati esistenti che eseguono già l'agente eliminato continuano a eseguire il check-in, i nodi che eseguono sistemi operativi Debian possono interrompere la creazione di report. Si consiglia di eseguire l'installazione manualmente `puppet-agent` su nodi che eseguono sistemi operativi per i quali il software dell'agente non è preinstallato sul server OpsWorks for Puppet Enterprise. Per informazioni dettagliate su come rendere `puppet-agent` disponibile sul server per i nodi con altri sistemi operativi, consulta Installazione di agenti nella documentazione Puppet Enterprise.

Per informazioni su come associare automaticamente i nodi al tuo Puppet master compilando i dati utente dell' EC2 istanza, consulta. Aggiungere nodi automaticamente in OpsWorks Puppet Enterprise

Eseguire chiamate API **`associateNode()`**

Dopo aver aggiunto i nodi mediante l'installazione`puppet-agent`, i nodi inviano le richieste di firma dei certificati (CSRs) al server OpsWorks for Puppet Enterprise. Puoi visualizzarli CSRs nella console Puppet; per maggiori informazioni sul node CSRs, consulta Gestire le richieste di firma dei certificati nella documentazione di Puppet Enterprise. L'esecuzione della chiamata `associateNode()` API OpsWorks for Puppet Enterprise elabora il nodo CSRs e associa il nodo al tuo server. Di seguito è riportato un esempio di come utilizzare questa chiamata API AWS CLI per associare un singolo nodo. È necessaria la richiesta CSR formattata PEM che il nodo invia; è possibile ottenerla dalla console Puppet.

```
aws opsworks-cm associate-node --server-name "test-puppet-
server" --node-name "node or instance ID" --engine-attributes
 "Name=PUPPET_NODE_CSR,Value='PEM_formatted_CSR_from_the_node'
```

Per ulteriori informazioni su come aggiungere nodi automaticamente utilizzando `associateNode()`, consulta Aggiungere nodi automaticamente in OpsWorks Puppet Enterprise.

Considerazioni per l'aggiunta di nodi locali

Dopo l'installazione `puppet-agent` sui computer locali o sulle macchine virtuali, puoi utilizzare uno dei due modi per associare i nodi locali al master di OpsWorks for Puppet Enterprise.

- Se un nodo supporta l'installazione di SDK AWS, AWS CLI o AWS Strumenti per PowerShell, è possibile utilizzare il metodo consigliato per l'associazione di un nodo, ovvero per eseguire una chiamata API `associateNode()`. Lo starter kit che scarichi quando crei per la prima volta un master OpsWorks for Puppet Enterprise mostra come assegnare ruoli ai nodi utilizzando i tag. È possibile applicare tag nello stesso momento in cui si stanno associando i nodi con il Puppet master specificando fatti affidabili nella richiesta CSR. Ad esempio, l'archivio di controllo demo incluso nello starter kit è configurato per utilizzare il tag per `pp_role` assegnare ruoli alle istanze Amazon. EC2 Per ulteriori informazioni su come aggiungere tag a una richiesta CSR come fatti affidabili, consulta Richieste di estensione (dati di certificato permanente) nella documentazione piattaforma Puppet.
- Se il nodo non è in grado di eseguire strumenti di AWS gestione o sviluppo, puoi comunque registrarlo nel tuo master OpsWorks for Puppet Enterprise nello stesso modo in cui lo registreresti con qualsiasi master Puppet Enterprise non gestito. Come indicato in questo argomento, l'installazione `puppet-agent` invia una CSR al master OpsWorks for Puppet Enterprise. Un utente Puppet autorizzato può firmare la CSR manualmente o configurare la firma automatica CSRs modificando il `autosign.conf` file archiviato nel master Puppet. Per ulteriori informazioni sulla configurazione della firma automatica e della modifica `autosign.conf`, consulta Configurazione SSL: firma automatica di richieste di certificati nella documentazione piattaforma Puppet.

Per associare i nodi locali a un Puppet master e consentire al Puppet master di accettare tutti CSRs, procedi come segue nella console Puppet Enterprise. Il parametro che controlla il comportamento è `puppet_enterprise::profile::master::allow_unauthenticated_ca`.

> **Important**
>
> Non è consigliabile abilitare il Puppet master in modo che accetti i file autofirmati CSRs o tutti i CSRs documenti autofirmati per motivi di sicurezza. Per impostazione predefinita, consentire l'autenticazione non autenticata CSRs rende un Puppet master accessibile al mondo. L'impostazione dell'abilitazione per impostazione predefinita del caricamento delle

richieste di certificati può rendere Puppet master vulnerabile ad attacchi Denial of Service (DoS).

1. Effettuare l'accesso alla console Puppet Enterprise.
2. Scegliere Configure (Configura), Classification (Classificazione) e PE Master, quindi selezionare la scheda Configuration (Configurazione).
3. Nella scheda Classification (Classificazione) individuare la classe puppet_enterprise::profile::master.
4. Imposta il valore del parametro allow_unauthenticated_ca su true.
5. Salvare le modifiche. Le modifiche vengono applicate durante la prossima esecuzione di Puppet. È possibile aspettare 30 minuti perché le modifiche diventino effettive (e i nodi locali vengano aggiunti) oppure avviare manualmente un'esecuzione Puppet nella sezione Run (Esegui) della console PE.

Ulteriori informazioni

Visita il sito del tutorial Learn Puppet per saperne di più sull'utilizzo OpsWorks dei server Puppet Enterprise e delle funzionalità della console Puppet Enterprise.

Effettuare l'accesso alla console Puppet Enterprise

Important

Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver scaricato le credenziali di accesso dalla pagina delle Proprietà del Puppet master e il server è online, accedere alla console Puppet Enterprise. In questo scenario, abbiamo istruito di specificare l'archivio di controllo che contiene i moduli e aggiungere almeno un nodo da gestire. In questo modo è possibile visualizzare le informazioni sull'agente e sui nodi nella console.

Quando tenti di connetterti alla pagina web della console di Puppet Enterprise, nel tuo browser vengono visualizzati avvisi relativi ai certificati finché non installi un certificato SSL AWS OpsWorks specifico e firmato da un'autorità di certificazione sul computer client che stai utilizzando per gestire il tuo server Puppet. Se non desideri visualizzare gli avvisi prima di continuare a navigare sulla pagina Web del pannello di controllo, installa il certificato SSL prima di effettuare l'accesso.

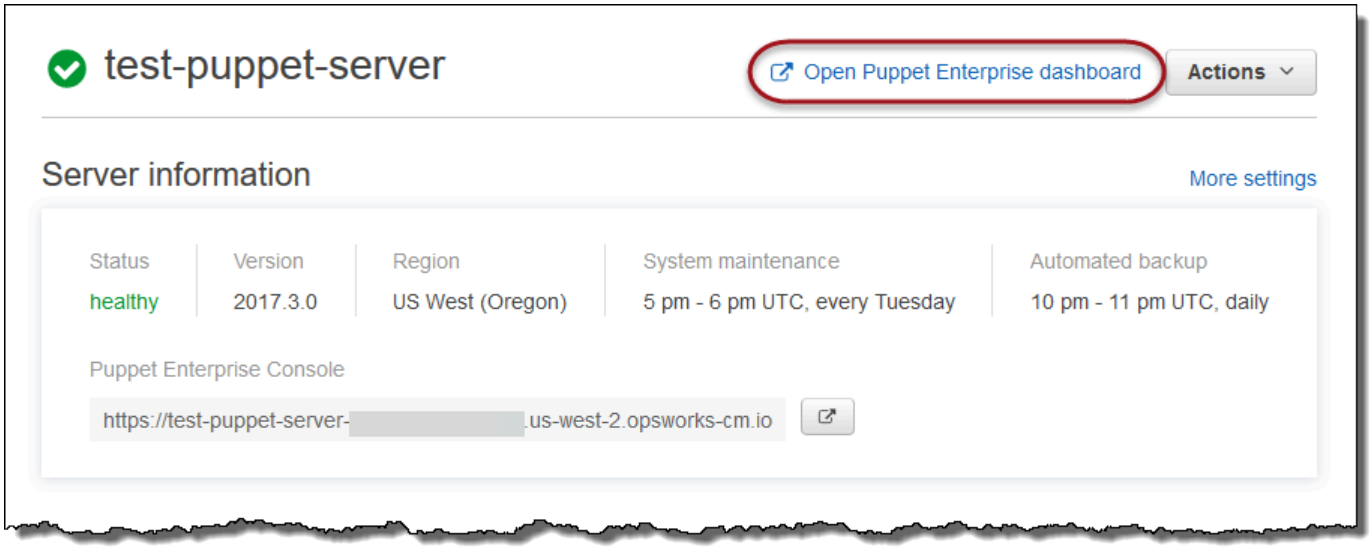
AWS OpsWorks Per installare il certificato SSL

- Scegli il certificato che corrisponde al tuo sistema.

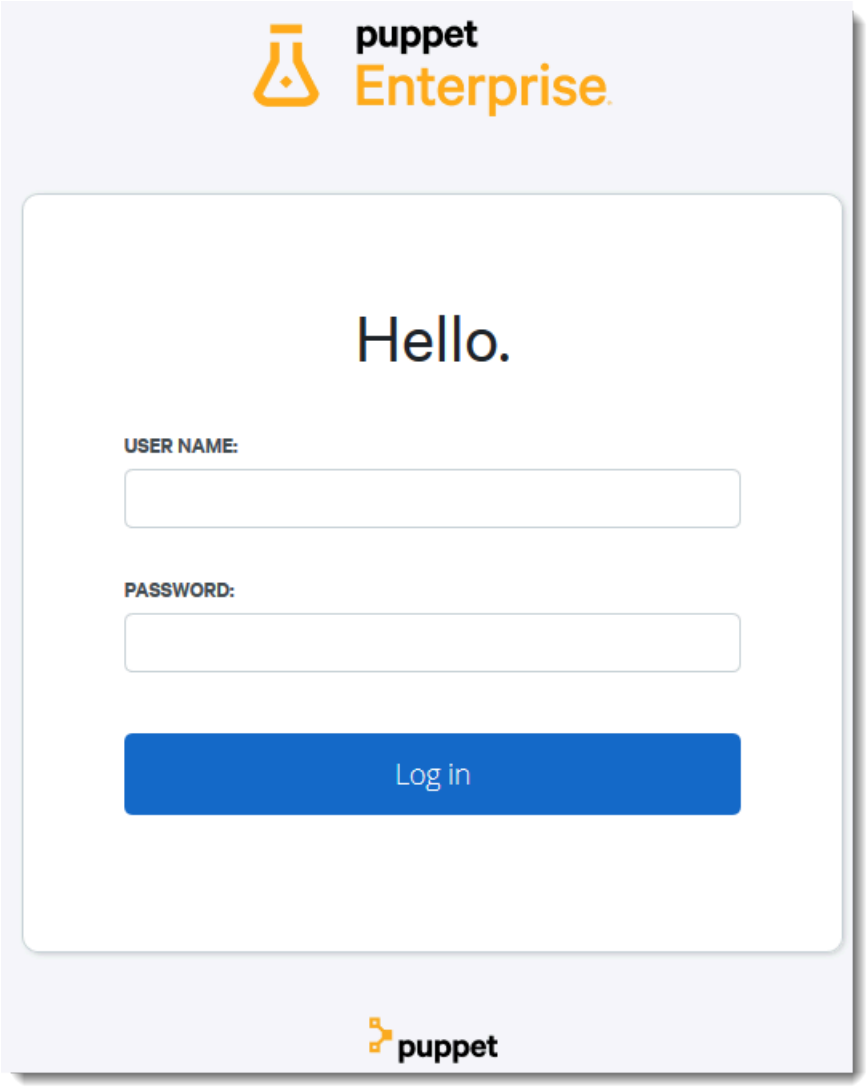
Dopo aver installato il certificato SSL sul lato client, puoi accedere alla console Puppet Enterprise senza visualizzare messaggi di avviso.

Per effettuare l'accesso alla console Puppet Enterprise

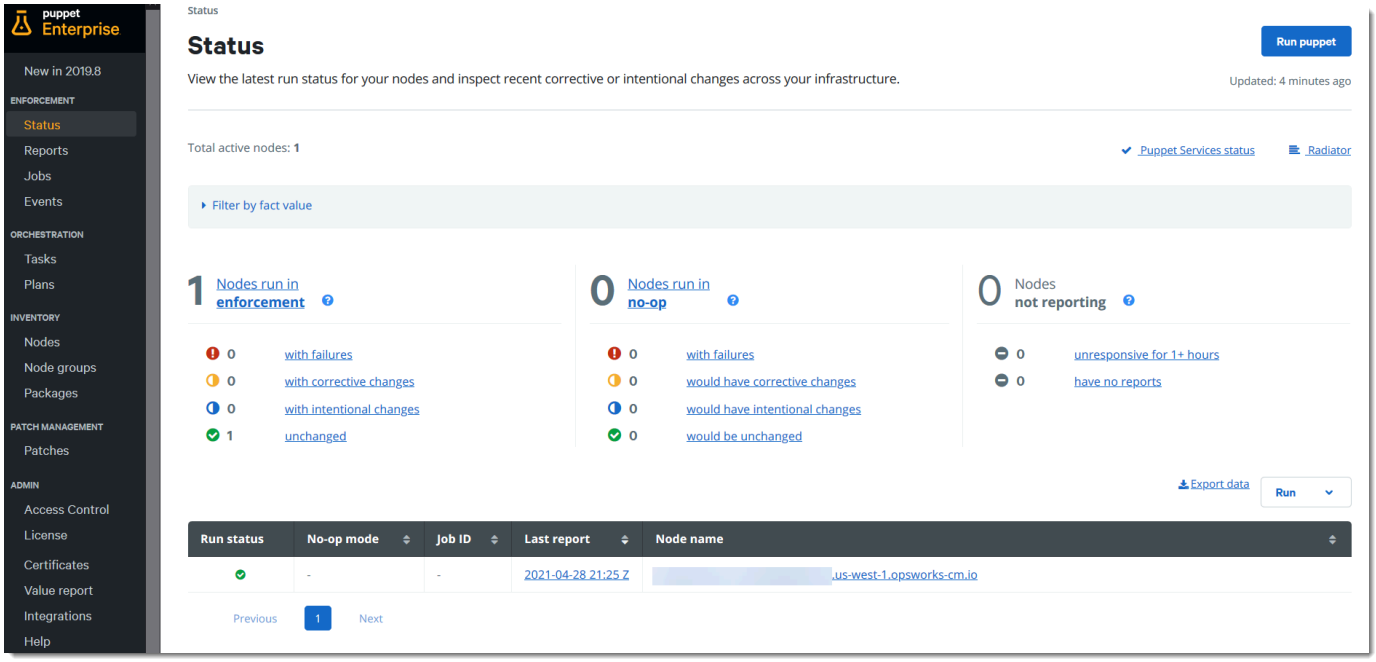
1. Decomprimere e aprire le credenziali di Puppet Enterprise scaricate in Prerequisiti. Devi disporre di queste credenziali per poter accedere.
2. In AWS Management Console, apri la pagina delle proprietà del tuo server Puppet.
3. In alto a destra della pagina Properties (Proprietà), selezionare Open Puppet Enterprise console (Apri console Puppet Enterprise).



4. Accedi utilizzando le credenziali della Fase 1.



5. Nella console Puppet Enterprise, è possibile visualizzare informazioni dettagliate sui nodi che si sta gestendo, sul progresso e sugli eventi dell'esecuzione del modulo, sul livello di conformità dei nodi e molto altro. Per ulteriori informazioni sulle funzionalità della console Puppet Enterprise e su come usarle, consulta Gestione dei nodi nella documentazione di Puppet Enterprise.



Raggruppare e classificare i nodi

Prima di specificare la configurazione desiderata dei nodi applicando classi, raggruppare i nodi in base ai loro ruoli all'interno dell'enterprise o alle loro caratteristiche comuni. Il raggruppamento e la classificazione dei nodi implica le seguenti attività di alto livello. È possibile completare queste attività utilizzando la console PE. Per informazioni dettagliate su come raggruppare e classificare i nodi, consulta Raggruppamento e classificazione dei nodi nella documentazione Puppet Enterprise.

1. Creazione di gruppi di nodi.
2. Aggiungere nodi a gruppi manualmente o automaticamente applicando le regole che si creano.
3. Assegnare classi a gruppi di nodi.

Reimpostazione della password dell'amministratore e dell'utente

Per informazioni su come modificare la password che usi per accedere alla console di Puppet Enterprise, vedi Reimpostare la password dell'amministratore della console nella documentazione di Puppet Enterprise.

Come impostazione predefinita, dopo dieci tentativi di accesso, gli utenti non possono più accedere alla console Puppet. Per ulteriori informazioni su come reimpostare le password degli utenti nel caso non possano più effettuare l'accesso, consulta Endpoint password nella documentazione Puppet Enterprise.

Facoltativo: utilizzalo AWS CodeCommit come archivio di controllo remoto Puppet r10k

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

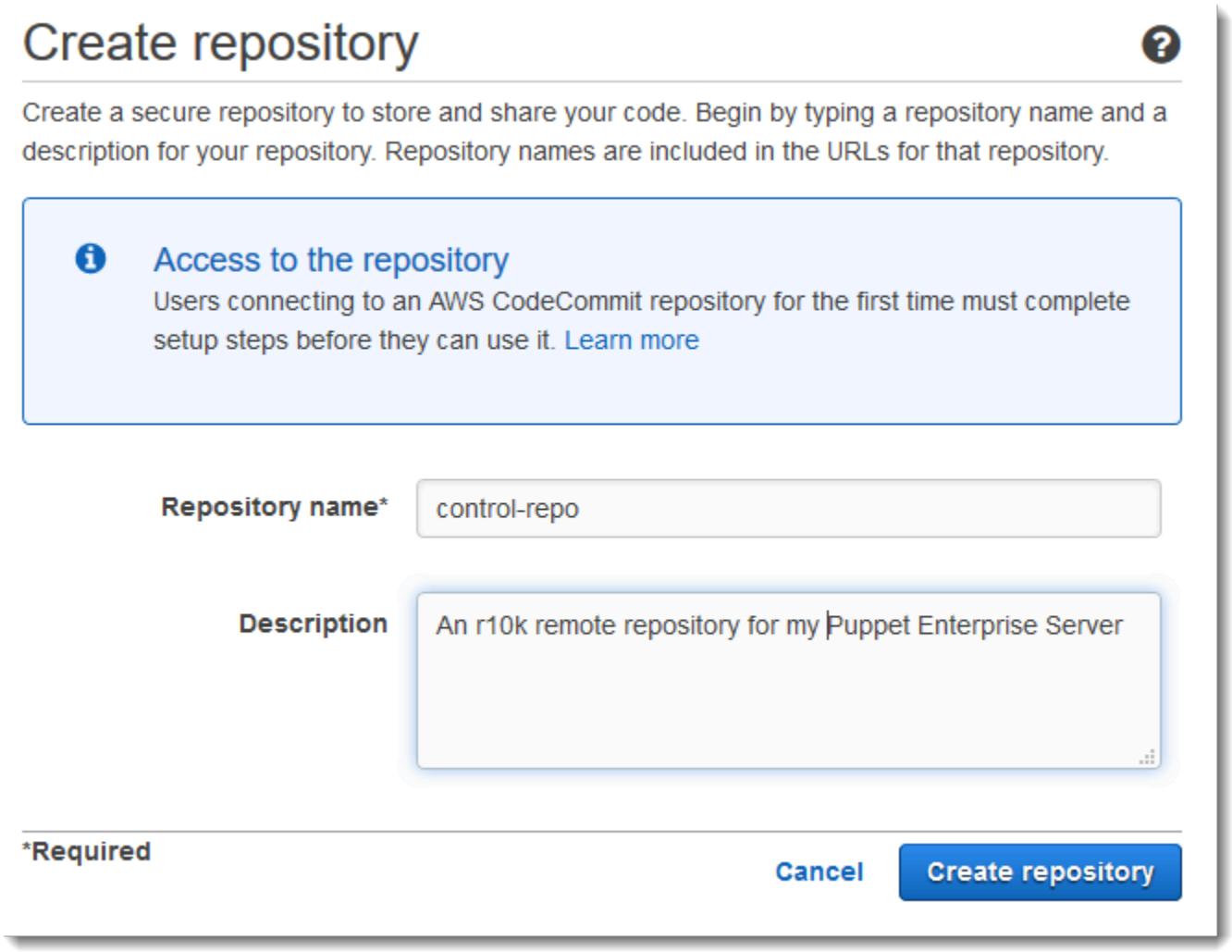
È possibile creare un nuovo repository utilizzando AWS CodeCommit e utilizzarlo come repository di controllo remoto r10k. Per completare i passaggi di questa sezione e utilizzare un CodeCommit repository, è necessario un utente che disponga delle autorizzazioni fornite dalla politica gestita. AWSCodeCommitReadOnly

Argomenti

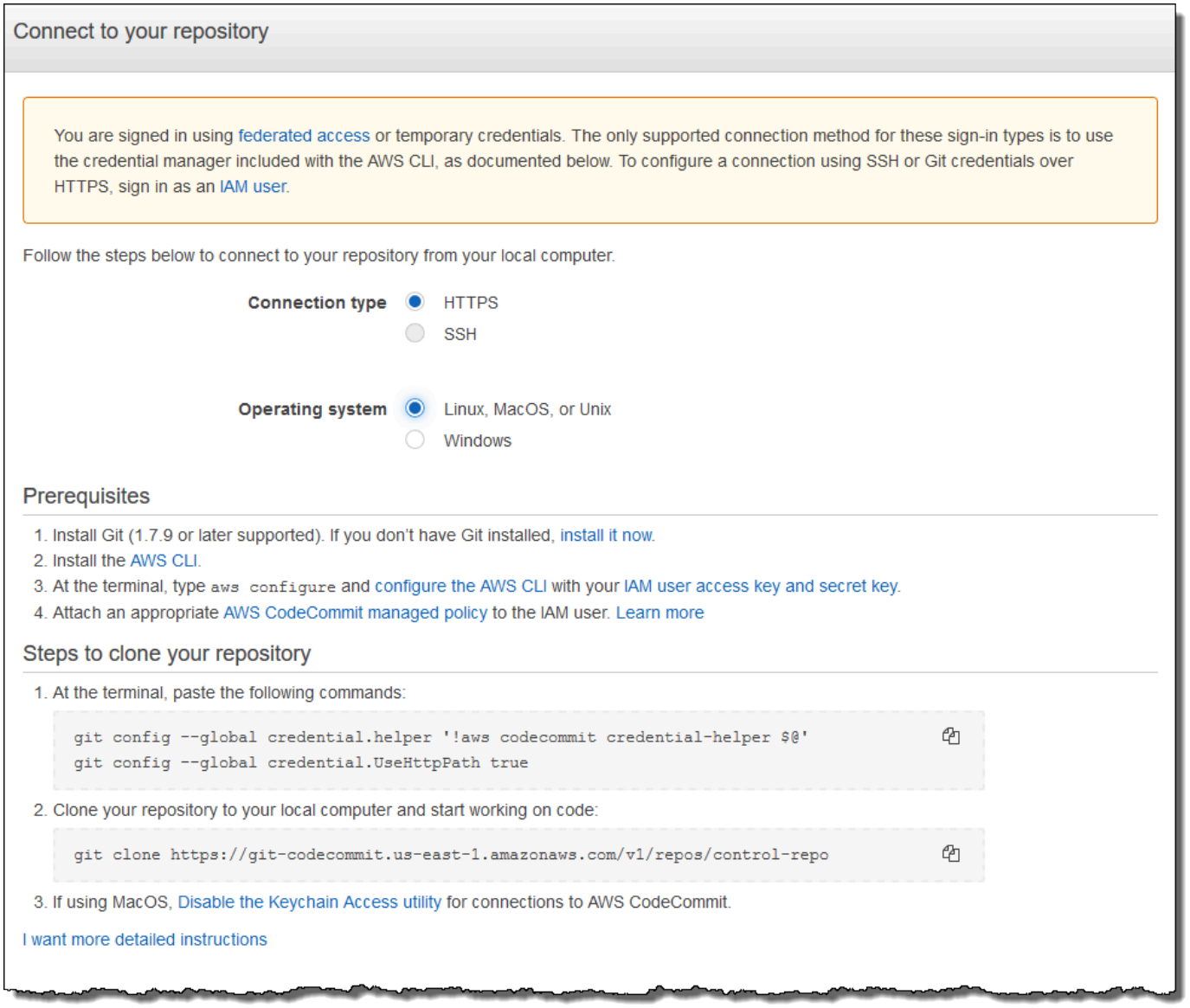
- Passaggio 1: utilizzare CodeCommit come repository con un tipo di connessione HTTPS
- Passaggio 2: (Facoltativo) Utilizzo CodeCommit come repository con un tipo di connessione SSH

Passaggio 1: utilizzare CodeCommit come repository con un tipo di connessione HTTPS

1. Nella CodeCommit console, crea un nuovo repository.



2. Scegli Salta per saltare la configurazione di un argomento di Amazon SNS.
3. Nella pagina Code (Codice), selezionare Connect to your repository (Collegarsi all'archivio).
4. Nella pagina Connect to your repository (Collegarsi all'archivio), selezionare HTTPS come Connection type (Tipo di connessione) e selezionare il sistema operativo.



Nell'area Steps to clone your repository (Fasi per clonare l'archivio) l'URL `git clone` deve essere simile al seguente: `https://git-codecommit.`*`region`*`.amazonaws.com/v1/repos/control-repo`. Copiare questo URL in un posto favorevole per utilizzarlo nella configurazione del server Puppet.

5. Chiudi la pagina Connect to your repository e torna alla configurazione del server OpsWorks for Puppet Enterprise.
6. Incollare l'URL copiato nella Fase 4 nella casella di stringa r10k remote (r10k remoto) nella pagina Configure credentials (Configura credenziali) della procedura guidata del Puppet master. Lasciare vuota la casella r10k private key (chiave privata r10k). Terminare la creazione e l'avvio del Puppet master.

7. Nella console IAM, collega la AWSCodeCommitReadOnlypolicy al ruolo del profilo di istanza del tuo Puppet master. Per ulteriori informazioni su come aggiungere una policy di autorizzazione a un ruolo IAM, consulta Adding IAM identity permissions (console) nella IAM User Guide.
8. Segui i passaggi descritti in Configurazione per gli utenti HTTPS che utilizzano le credenziali Git nella Guida per l'AWS CodeCommit utente per inviare i `control-repo` contenuti esistenti al nuovo CodeCommit repository.
9. Ora è possibile continuare seguendo le istruzioni the section called “Finire la configurazione” e utilizzare lo Starter Kit per distribuire il codice al Puppet master. Il comando seguente è un esempio.

```
puppet-code deploy --all --wait --config-file .config/puppet-code.conf
```

Passaggio 2: (Facoltativo) Utilizzo CodeCommit come repository con un tipo di connessione SSH

È possibile configurare un repository di controllo remoto AWS CodeCommit r10k per utilizzare l'autenticazione tramite key pair SSH. Prima di iniziare questa procedura, devono essere stati completati i seguenti prerequisiti.

- È necessario aver avviato il server OpsWorks for Puppet Enterprise con un repository di controllo HTTPS come descritto nella sezione precedente,. the section called “Passaggio 1: utilizzare CodeCommit come repository con un tipo di connessione HTTPS” Questo requisito deve essere completato prima in modo da poter caricare la configurazione richiesta nel Puppet master.
- Assicurati di avere un utente con la policy AWSCodeCommitReadOnlygestita allegata. Per ulteriori informazioni su come creare un utente, consulta Creating an IAM user in your AWS account nella IAM User Guide.
- Creare una chiave SSH e associarla all'utente . Segui le istruzioni per creare una public/private key pair *`ssh-keygen`* nel Passaggio 3: Configurazione delle credenziali su Linux, macOS o Unix nella AWS CodeCommit Guida per l'utente.

1. In una AWS CLI sessione, esegui il comando seguente per caricare il contenuto del file della chiave privata in Parameter Store. AWS Systems Manager Il server OpsWorks for Puppet Enterprise esegue una query su questo parametro per ottenere un file di certificato richiesto. *`private_key_file`*Sostituiscilo con il percorso del tuo file di chiave privata SSH.

```
aws ssm put-parameter --name puppet_user_pk --type String --value
 "`cat private_key_file`"
```

2. Aggiungi le autorizzazioni di Systems Manager Parameter Store al tuo Puppet master.

   a. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

   b. Nel pannello di navigazione a sinistra, seleziona Ruoli.

   c. Scegli 2 ruoliaws-opsworks-cm-ec.

   d. Nella scheda Autorizzazioni, scegli Collega policy.

   e. Nella barra Cerca immettere **`AmazonSSMManagedInstanceCore`**.

   f. Nei risultati della ricerca, scegli Amazon SSMManaged InstanceCore.

   g. Scegli Collega policy.

3. Creare il manifest del file di configurazione. Se si sta usando il repository `control-repo-example` fornito nello starter kit, creare i file seguenti nei percorsi mostrati nel repository di esempio. In alternativa, crearli in base alla struttura del repository di controllo. Sostituisci il *IAM_USER_SSH_KEY* valore con l'ID della chiave SSH che hai creato nei prerequisiti per questa procedura.

```
control-repo-example/site/profile/manifests/codecommit.pp
```

```
class profile::codecommit {
  $configfile = @(CONFIGFILE)
      Host git-codecommit.*.amazonaws.com
      User IAM_USER_SSH_KEY
      IdentityFile /etc/puppetlabs/puppetserver/ssh/codecommit.rsa
      StrictHostKeyChecking=no
      | CONFIGFILE

  # Replace REGION with the correct region for your server.
  $command = @(COMMAND)
      aws ssm get-parameters \
      --region REGION \
      --names puppet_user_pk \
      --query "Parameters[0].Value" \
      --output text >| /etc/puppetlabs/puppetserver/ssh/codecommit.rsa
      | COMMAND

  $dirs = [
```

```
            '/opt/puppetlabs/server/data/puppetserver/.ssh',
            '/etc/puppetlabs/puppetserver/ssh',
          ]

  file { $dirs:
    ensure => 'directory',
    group  => 'pe-puppet',
    owner  => 'pe-puppet',
    mode   => '0750',
  }

  file { 'ssh-config':
    path    => '/opt/puppetlabs/server/data/puppetserver/.ssh/config',
    require => File[$dirs],
    content => $configfile,
    group   => 'pe-puppet',
    owner   => 'pe-puppet',
    mode    => '0600',
  }

  exec { 'download-codecommit-certificate':
    command => $command,
    require => File[$dirs],
    creates => '/etc/puppetlabs/puppetserver/ssh/codecommit.rsa',
    path    => '/bin',
    cwd     => '/etc/puppetlabs',
  }

  file { 'private-key-permissions':
    subscribe => Exec['download-codecommit-certificate'],
    path      => '/etc/puppetlabs/puppetserver/ssh/codecommit.rsa',
    group     => 'pe-puppet',
    owner     => 'pe-puppet',
    mode      => '0600',
  }
}
```

4. Invia il tuo repository di controllo a. CodeCommit Eseguire i comandi seguenti per inviare il nuovo file manifest al repository.

```
git add ./site/profile/manifests/codecommit.pp
git commit -m 'Configuring for SSH connection to CodeCommit'
git push origin production
```

5. Distribuire i file manifest. Esegui i seguenti comandi per distribuire la configurazione aggiornata sul tuo server OpsWorks for Puppet Enterprise. Sostituisci *STARTER_KIT_DIRECTORY* con il percorso dei file di configurazione di Puppet.

```
cd STARTER_KIT_DIRECTORY

puppet-access login --config-file .config/puppetlabs/client-tools/puppet-
access.conf

puppet-code deploy --all --wait \
--config-file .config/puppet-code.conf \
--token-file .config/puppetlabs/token
```

6. Aggiorna OpsWorks la classificazione del server Puppet Enterprise. Per impostazione predefinita, l'agente Puppet viene eseguito sui nodi (tra cui il master) ogni 30 minuti. Per evitare l'attesa, è possibile eseguire manualmente l'agente sul Puppet master. L'esecuzione dell'agente preleva il nuovo file manifest.

   a. Effettuare l'accesso alla console Puppet Enterprise.

   b. Scegliere Classification (Classificazione).

   c. Espandere PE Infrastructure (Infrastruttura PE).

   d. Scegliere PE Master (Master PE).

   e. Nella scheda Configuration (Configurazione) immettere **`profile::codecommit`** in Add new class (Aggiungi nuova classe).

      La nuova classe, `profile::codecommit`, potrebbe non essere visualizzata immediatamente dopo l'esecuzione di `puppet-code deploy`. Scegliere Refresh (Aggiorna) in questa pagina se non viene visualizzata.

   f. Scegliere Add class (Aggiungi classe), quindi scegliere Commit 1 change (Modifica commit 1).

   g. Esegui manualmente l'agente Puppet sul server OpsWorks for Puppet Enterprise. Scegliere Nodes (Nodi), scegliere il server nell'elenco, scegliere Run Puppet (Esegui Puppet), quindi scegliere Run (Esegui).

7. Nella console Puppet Enterprise modificare l'URL del repository per l'utilizzo di SSH anziché di HTTPS. La configurazione eseguita in questi passaggi viene salvata durante il processo di backup e ripristino di OpsWorks for Puppet Enterprise, quindi non è necessario modificare manualmente la configurazione del repository dopo le attività di manutenzione.

a. Scegliere Classification (Classificazione).
b. Espandere PE Infrastructure (Infrastruttura PE).
c. Scegliere PE Master (Master PE).
d. Nella scheda Configuration (Configurazione) individuare la classe `puppet_enterprise::profile::master`.
e. Scegliere Edit (Modifica) accanto al parametro `r10k_remote`.
f. Sostituire l'URL HTTPS con l'URL SSH per il repository e scegliere Commit 1 change (Modifica commit 1).
g. Esegui manualmente l'agente Puppet sul server OpsWorks for Puppet Enterprise. Scegliere Nodes (Nodi), scegliere il server nell'elenco, scegliere Run Puppet (Esegui Puppet), quindi scegliere Run (Esegui).

Crea un AWS OpsWorks for Puppet Enterprise master utilizzando AWS CloudFormation

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks for Puppet Enterprise ti consente di eseguire un server Puppet Enterprise in. AWSÈ possibile effettuare il provisioning di un server master Puppet Enterprise in circa 15 minuti.

A partire dal 3 maggio 2021, OpsWorks for Puppet Enterprise memorizza alcuni attributi del server Puppet Enterprise in. AWS Secrets Manager Per ulteriori informazioni, consulta Integrazione con AWS Secrets Manager.

La procedura dettagliata seguente ti aiuta a creare un Puppet master in OpsWorks per Puppet Enterprise creando uno stack in. AWS CloudFormation

Argomenti

- Prerequisiti
- Creare un Puppet Enterprise Master in AWS CloudFormation

Prerequisiti

Prima di creare un nuovo Puppet master, crea le risorse esterne a Puppet Enterprise necessarie OpsWorks per accedere e gestire il tuo Puppet master. Per ulteriori informazioni, consulta Prerequisiti nella sezione delle nozioni di base di questa guida.

Se si sta creando un server che utilizza un dominio personalizzato, è necessario un dominio, un certificato e una chiave privata personalizzati. Devi specificare i valori per tutti e tre questi parametri nel tuo modello. AWS CloudFormation Per ulteriori informazioni sui requisiti per i `CustomPrivateKey` parametri, e`CustomDomain`,,`CustomCertificate`,,,,,,, CreateServerconsulta il AWS OpsWorks CM API Reference.

Consulta la sezione OpsWorks -CM della Guida per AWS CloudFormation l'utente di riferimento per conoscere i valori supportati e obbligatori nel AWS CloudFormation modello utilizzato per creare il server.

Creare un Puppet Enterprise Master in AWS CloudFormation

Questa sezione descrive come utilizzare un AWS CloudFormation modello per creare uno stack che crei un server master OpsWorks for Puppet Enterprise. Puoi farlo utilizzando la AWS CloudFormation console o il. AWS CLIÈ disponibile un AWS CloudFormation modello di esempio da utilizzare per creare uno stack di server OpsWorks per Puppet Enterprise. Assicurati di aggiornare il modello di esempio con il nome del server, i ruoli IAM, il profilo dell'istanza, la descrizione del server, il numero di conservazione dei backup, le opzioni di manutenzione e i tag opzionali. Se il server utilizzerà un dominio personalizzato, è necessario specificare i valori per i parametri `CustomDomain`, `CustomCertificate` e `CustomPrivateKey` nel modello AWS CloudFormation . Per ulteriori informazioni su tali opzioni, consulta the section called “Crea un Puppet Enterprise Master utilizzando il AWS Management Console” nella sezione delle nozioni di base di questa guida.

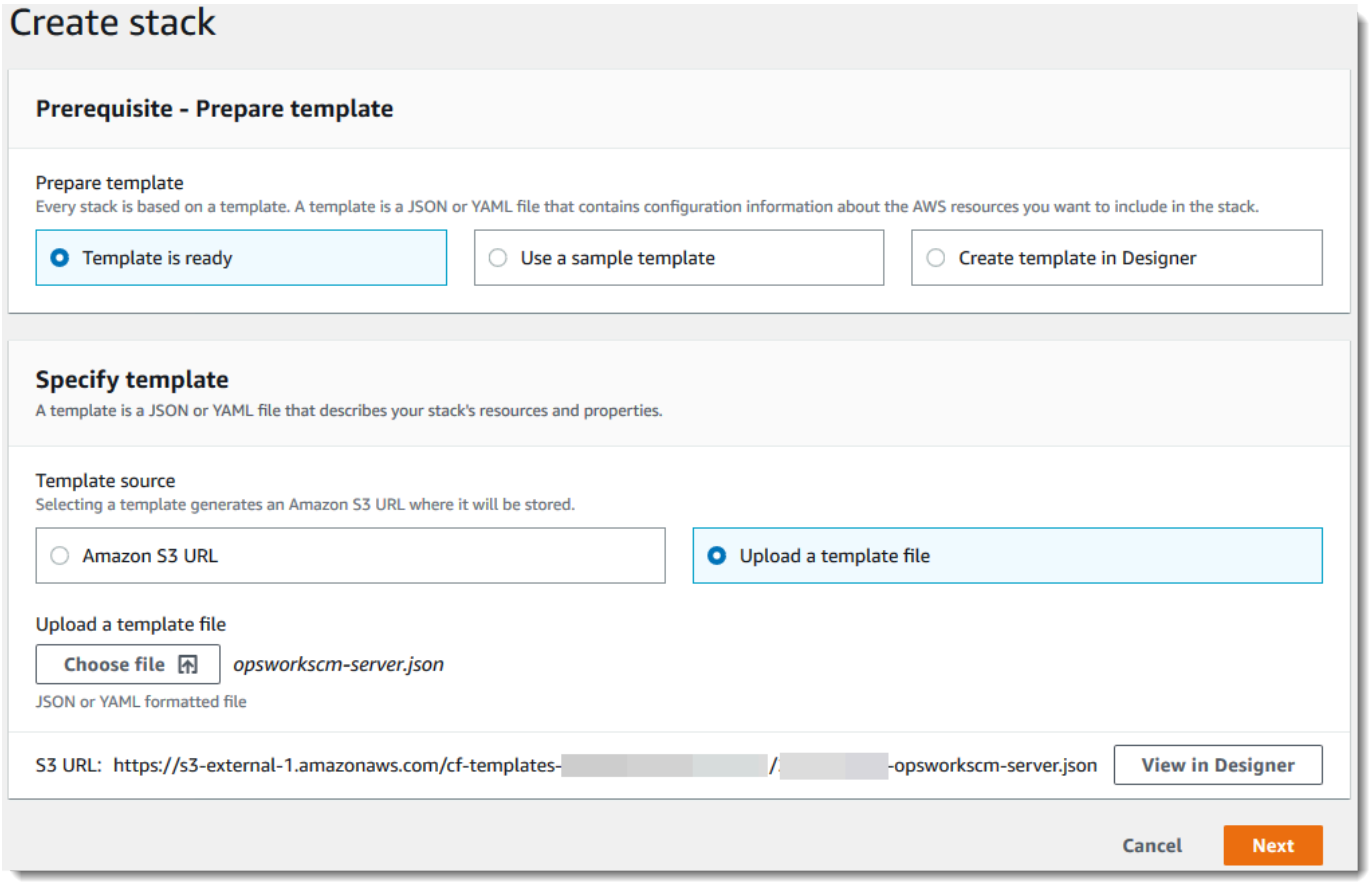
Argomenti

- Crea un Puppet Enterprise Master utilizzando AWS CloudFormation (Console)
- Crea un Puppet Enterprise Master utilizzando AWS CloudFormation (CLI)

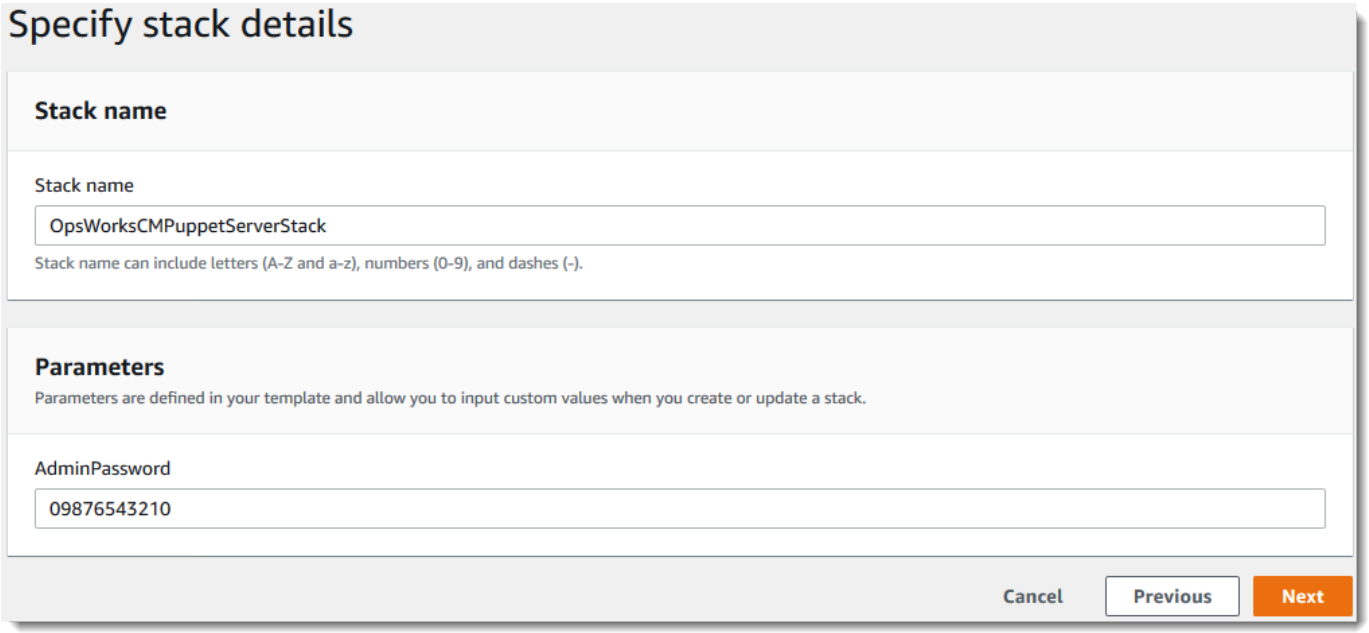
Crea un Puppet Enterprise Master utilizzando AWS CloudFormation (Console)

1. Accedi AWS Management Console e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformazione.
2. Nella AWS CloudFormation home page, scegli Crea stack.
3. In Prerequisite - Prepare template (Prerequisito - Preparare modello), se si sta utilizzando il modello AWS CloudFormation di esempio, scegliere Template is ready (Modello pronto).
4. In Specify template (Specifica modello), scegliere l'origine del modello. Per questa procedura dettagliata, scegli Carica un file modello e carica un AWS CloudFormation modello che crei un server Puppet Enterprise. Cercare il file di modello e quindi scegliere Avanti.

   Un AWS CloudFormation modello può essere in formato YAML o JSON. È disponibile un AWS CloudFormation modello di esempio da utilizzare; assicurati di sostituire i valori di esempio con i tuoi. Puoi utilizzare lo strumento di progettazione del AWS CloudFormation modello per creare un nuovo modello o convalidarne uno esistente. Per ulteriori informazioni su come eseguire questa operazione, consulta la sezione relativa alla panoramica dell'interfaccia di progettazione AWS CloudFormation nella Guida per l'utente AWS CloudFormation .



5. Nella pagina Specify stack details (Specifica dettagli dello stack), immettere un nome per lo stack. Questo sarà diverso dal nome del server, è solo un nome di stack. Nell'area Parameters (Parametri), immettere una password di amministratore per l'accesso alla pagina Web della console Puppet Enterprise. La password deve utilizzare da 8 a 32 caratteri ASCII. Scegli Next (Successivo).



6. Nella pagina Opzioni, puoi aggiungere tag al server che stai creando con lo stack e scegliere un ruolo IAM per la creazione di risorse se non hai già specificato un ruolo IAM da utilizzare nel tuo modello. Dopo aver specificato le opzioni, scegliere Next (Avanti). Per ulteriori informazioni sulle opzioni avanzate come i trigger di rollback, consulta Setting AWS CloudFormation Stack Options nella Guida per l'utente.AWS CloudFormation

7. Nella pagina Review (Esamina), esaminare le proprie opzioni. Quando si è pronti per creare lo stack del server, scegliere Create (Crea).

   Mentre aspetti di AWS CloudFormation creare lo stack, visualizza lo stato di creazione dello stack. Se si verifica un errore di creazione dello stack, esaminare i messaggi di errore visualizzati nella console per risolvere i problemi. Per ulteriori informazioni sulla risoluzione dei problemi relativi a errori dello stack AWS CloudFormation , consulta la sezione relativa alla risoluzione dei problemi relativi agli errori nella Guida per l'utente AWS CloudFormation .

   Al termine della creazione del server, il Puppet master è disponibile nella home page OpsWorks di Puppet Enterprise, con lo stato online. Dopo che il server è online, la console Puppet Enterprise è disponibile nel dominio del server, in un URL nel formato seguente: `https://`*`your_server_name`*`-`*`randomID`*`.`*`region`*`.opsworks-cm.io`.

> **Note**
>
> Se hai specificato un dominio, un certificato e una chiave privata personalizzati per il tuo server, crea una voce CNAME nello strumento di gestione DNS dell'azienda che mappa il dominio personalizzato all'endpoint generato automaticamente OpsWorks per Puppet Enterprise per il server. Non è possibile gestire il server o connettersi al sito Web di gestione Puppet Enterprise per il server finché non si esegue il mapping dell'endpoint generato al valore del dominio personalizzato.
> Per ottenere il valore dell'endpoint generato, esegui il AWS CLI comando seguente dopo che il server è online:
>
> ```
> aws opsworks describe-servers --server-name server_name
> ```

Crea un Puppet Enterprise Master utilizzando AWS CloudFormation (CLI)

Se sul tuo computer locale non è già in esecuzione AWS CLI, scaricalo e installalo AWS CLI seguendo le istruzioni di installazione nella AWS Command Line Interface User Guide. Questa sezione non descrive tutti i parametri che è puoi utilizzare con il comando create-stack. Per ulteriori informazioni sui parametri create-stack, consulta create-stack nella documentazione di riferimento delle AWS CLI .

1. Assicurati di completare il modulo Prerequisiti per creare un master OpsWorks for Puppet Enterprise.
2. Crea un ruolo di servizio e un profilo di istanza.

   Trova e copia i ruoli ARNs di servizio nel tuo account.

   ```
   aws iam list-roles --path-prefix "/service-role/" --no-paginate
   ```

   Nei risultati del comando `list-roles`, cercare le voci del ruolo di servizio e del profilo dell'istanza che sono simili a quanto segue. Prendi nota del ruolo di servizio e ARNs del profilo dell'istanza e aggiungili al AWS CloudFormation modello che stai utilizzando per creare lo stack del server master Puppet.

   ```
   {
       "AssumeRolePolicyDocument": {
   ```

```
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "ec2.amazonaws.com"
                }
            }
        ]
    },
    "RoleId": "AROZZZZZZZZZZQG6R22HC",
    "CreateDate": "2018-01-05T20:42:20Z",
    "RoleName": "aws-opsworks-cm-ec2-role",
    "Path": "/service-role/",
    "Arn": "arn:aws:iam::000000000000:role/service-role/aws-opsworks-cm-ec2-role"
},
{
    "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "opsworks-cm.amazonaws.com"
                }
            }
        ]
    },
    "RoleId": "AROZZZZZZZZZZZZZZZ6QE",
    "CreateDate": "2018-01-05T20:42:20Z",
    "RoleName": "aws-opsworks-cm-service-role",
    "Path": "/service-role/",
    "Arn": "arn:aws:iam::000000000000:role/service-role/aws-opsworks-cm-service-
role"
}
```

3. Crea il master OpsWorks for Puppet Enterprise eseguendo nuovamente il create-stack comando.

   - *stack_name*Sostituiscilo con il nome del tuo stack. Questo è il nome dello AWS CloudFormation stack, non del tuo Puppet master. Il nome del master Puppet è il valore `ServerName` nel modello AWS CloudFormation .

- Sostituiscilo *`template`* con il percorso del file modello e l'estensione *`yaml or json`* con `.yaml` o, `.json` se necessario.
- I valori di `--parameters` corrispondono a EngineAttributesdall'CreateServerAPI. Per Puppet, i seguenti attributi del motore sono forniti dall'utente per la creazione di un server. Gli attributi del motore r10k collegano il master Puppet a un repository di codice per la gestione della configurazione dell'ambiente del server. Per ulteriori informazioni sugli attributi del motore r10k, consulta la sezione relativa alla gestione di codice con r10k nella documentazione di Puppet Enterprise.
  - `PUPPET_ADMIN_PASSWORD`, una password di amministratore per l'accesso alla pagina Web della console Puppet Enterprise. La password deve contenere da 8 a 32 caratteri ASCII e includere almeno una lettera maiuscola, una lettera minuscola, un numero e un carattere speciale.
  - `PUPPET_R10K_REMOTE`, l'URL del repository di controllo (ad esempio, ssh://git@your.git-repo.com:user/control-repo.git). Specificare un r10k remoto apre la porta TCP 8170.
  - `PUPPET_R10K_PRIVATE_KEY`. Se stai usando un repository Git privato, aggiungi PUPPET_R10K_PRIVATE_KEY per specificare un URL SSH e una chiave SSH privata con codifica PEM.

```
aws cloudformation create-stack --stack-name stack_name
 --template-body file://template.yaml or json --parameters
 ParameterKey=AdminPassword,ParameterValue="password"
```

Di seguito è riportato un esempio.

```
aws cloudformation create-stack --stack-name "OpsWorksCMPuppetServerStack"
 --template-body file://opsworkscm-puppet-server.json --parameters
 ParameterKey=AdminPassword,ParameterValue="09876543210Ab#"
```

L'esempio seguente specifica gli attributi del motore r10k come parametri, quando non sono forniti nel modello. AWS CloudFormation Un modello di esempio che include gli attributi del motore r10k, `puppet-server-param-attributes.yaml`, è incluso nei modelli AWS CloudFormation di esempio.

```
aws cloudformation create-stack --stack-name MyPuppetStack --
template-body file://puppet-server-param-attributes.yaml --parameters
 ParameterKey=AdminPassword,ParameterValue="superSecret1%3"
```

```
ParameterKey=R10KRemote,ParameterValue="https://www.yourRemote.com"
ParameterKey=R10KKey,ParameterValue="$(cat puppet-r10k.pem)"
```

L'esempio seguente specifica gli attributi del motore r10k e i relativi valori nel modello AWS CloudFormation ; il comando deve solo puntare al file del modello. Il modello specificato come valore di `--template-body`, `puppet-server-in-file-attributes.yaml`, è incluso nel modelli AWS CloudFormation di esempio.

```
aws cloudformation create-stack --stack-name MyPuppetStack --template-body file://
puppet-server-in-file-attributes.yaml
```

4. (Facoltativo) Per ottenere lo stato di creazione dello stack, eseguire il comando seguente.

```
aws cloudformation describe-stacks --stack-name stack_name
```

5. Al termine della creazione dello stack, passare alla sezione successiva, the section called “Finire la configurazione”. Se si verifica un errore di creazione dello stack, esaminare i messaggi di errore visualizzati nella console per risolvere i problemi. Per ulteriori informazioni sulla risoluzione degli errori negli AWS CloudFormation stack, vedere Risoluzione dei problemi degli errori nella Guida per l'utente.AWS CloudFormation

Aggiorna un server OpsWorks for Puppet Enterprise per utilizzare un dominio personalizzato

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione descrive come aggiornare un server OpsWorks for Puppet Enterprise esistente per utilizzare un dominio e un certificato personalizzati utilizzando un backup del server per creare un nuovo server. In sostanza, stai copiando un server OpsWorks for Puppet Enterprise 2.0 esistente

creando un nuovo server da un backup, quindi configurando il nuovo server per utilizzare un dominio, un certificato e una chiave privata personalizzati.

Argomenti

- Prerequisiti
- Limitazioni
- Aggiornare un server per utilizzare un dominio personalizzato
- Vedi anche

Prerequisiti

Di seguito sono riportati i requisiti per aggiornare un server OpsWorks for Puppet Enterprise esistente per utilizzare un dominio e un certificato personalizzati.

- Il server che desideri aggiornare (o copiare) deve eseguire Puppet Enterprise 2019.8.5.
- Decidere quale backup si desidera utilizzare per creare un nuovo server. È necessario avere almeno un backup disponibile del server che si desidera aggiornare. Per ulteriori informazioni sui backup in Puppet Enterprise, consulta OpsWorks . Esegui il backup di un OpsWorks Puppet Enterprise Server
- Tieni a portata di mano il ruolo di servizio e il profilo di istanza ARNs che hai usato per creare il server esistente che è l'origine del backup.
- Assicurarsi di utilizzare la versione più recente della AWS CLI. Per ulteriori informazioni sull'aggiornamento AWS CLI degli strumenti, consulta la Guida per l'utente dell'interfaccia AWS CLI a riga di comando di AWS (Installazione della Guida per l'utente dell'interfaccia a riga di comando di AWS).

Limitazioni

Quando aggiorni un server esistente creando un nuovo server da un backup, il nuovo server non può essere esattamente lo stesso del server esistente OpsWorks per Puppet Enterprise.

- È possibile completare questa procedura solo utilizzando AWS CLI o uno dei AWS SDKs. Non è possibile creare un nuovo server da un backup utilizzando la AWS Management Console.

- Il nuovo server non può utilizzare lo stesso nome del server esistente all'interno di un account e all'interno di una regione AWS. Il nome deve essere diverso da quello del server esistente utilizzato come origine del backup.
- I nodi collegati al server esistente non vengono gestiti dal nuovo server. È necessario effettuare una delle seguenti operazioni.
  - Collegare nodi diversi, perché i nodi non possono essere gestiti da più master Puppet.
  - Eseguire la migrazione dei nodi dal server esistente (l'origine del backup) al nuovo server e al nuovo endpoint di dominio personalizzato. Per ulteriori informazioni sulla migrazione dei nodi, vedere la documentazione di Puppet Enterprise.

Aggiornare un server per utilizzare un dominio personalizzato

Per aggiornare un master Puppet esistente, è possibile crearne una copia eseguendo il comando `create-server`, aggiungendo parametri per specificare un backup, un dominio personalizzato, un certificato personalizzato e una chiave privata personalizzata.

1. Se non disponi di un ruolo di servizio o di un profilo ARNs di istanza da specificare nel `create-server` comando, segui i passaggi da 1 Crea un server Chef Automate utilizzando il AWS CLI a 5 per creare un ruolo di servizio e un profilo di istanza da utilizzare.
2. Se non è stato già fatto, trovare il backup del master Puppet esistente su cui si desidera basare un nuovo server con un dominio personalizzato. Esegui il comando seguente per mostrare le informazioni su tutti i OpsWorks backup di Puppet Enterprise nel tuo account e in una regione. Assicurarsi di annotare l'ID del backup che si desidera utilizzare.

   ```
   aws opsworks-cm --region region name describe-backups
   ```

3. Crea il server OpsWorks for Puppet Enterprise eseguendo il comando. `create-server`

   - Il `--engine` valore è `PuppetMonolithic`, `--engine-model` è ed `--engine-version` è `2019` o`2017`.
   - Il nome del server deve essere univoco all'interno del tuo AWS account, all'interno di ogni regione. I nomi dei server devono iniziare con una lettera; quindi sono consentiti lettere, numeri o trattini (-) fino a un massimo di 40 caratteri.
   - Utilizza l'ARN del profilo dell'istanza e l'ARN del ruolo di servizio copiati nella Fase 3 e 4.

- I tipi di istanza validi sono `c4.large`, `c4.xlarge` o `c4.2xlarge`. Per ulteriori informazioni sulle specifiche di questi tipi di istanze, consulta Instance Types nella Amazon EC2 User Guide.
- Il parametro `--engine-attributes` è opzionale; se non si specifica una password di amministratore Puppet, il processo di creazione del server ne genera uno. Se si aggiunge `--engine-attributes`, specificare una `PUPPET_ADMIN_PASSWORD`, una password di amministratore per l'accesso alla pagina Web della console Puppet Enterprise. La password deve utilizzare da 8 a 32 caratteri ASCII.
- Una coppia di chiavi SSH è facoltativa, ma può aiutare a connettersi al Puppet master se si deve reimpostare la password di amministratore della console. Per ulteriori informazioni sulla creazione di una coppia di chiavi SSH, consulta Amazon EC2 Key Pairs nella Amazon EC2 User Guide.
- Per utilizzare un dominio personalizzato, aggiungere i seguenti parametri al comando. Diversamente, il processo di creazione del master Puppet genera automaticamente un endpoint. Tutti e tre i parametri sono necessari per configurare un dominio personalizzato. Per informazioni sui requisiti aggiuntivi per l'utilizzo di questi parametri, consulta CreateServeril AWS OpsWorks CM API Reference.
  - `--custom-domain` - Un endpoint pubblico opzionale di un server, ad esempio `https://aws.my-company.com`.
  - `--custom-certificate` - Un certificato HTTPS in formato PEM. Il valore può essere un singolo certificato autofirmato o una catena di certificati.
  - `--custom-private-key` - Una chiave privata in formato PEM per la connessione al server tramite HTTPS. La chiave privata non deve essere crittografata; non può essere protetta da password o da passphrase.
- La manutenzione del sistema settimanale è obbligatoria. I valori validi devono essere specificati nel seguente formato: `DDD:HH:MM`. L'ora specificata è espressa nel Tempo coordinato universale (UTC). Se non si specifica un valore per `--preferred-maintenance-window`, il valore predefinito è un periodo casuale di un'ora il martedì, mercoledì o venerdì.
- I valori validi per `--preferred-backup-window` devono essere specificati in uno dei seguenti formati: `HH:MM` per backup giornaliero o `DDD:HH:MM` per backup settimanali. L'orario specificato è nel formato UTC. Il valore predefinito è un'ora di avvio giornaliera casuale. Per disattivare i backup automatici, aggiungere invece il parametro `--disable-automated-backup`.

- Per `--security-group-ids`, inserisci uno o più gruppi di sicurezza IDs, separati da uno spazio.
- Per `--subnet-ids`, inserire un ID di sottorete.

```
aws opsworks-cm create-server --engine "Puppet" --engine-model "Monolithic"
 --engine-version "2019" --server-name "server_name" --instance-profile-arn
 "instance_profile_ARN" --instance-type "instance_type" --engine-attributes
 '{"PUPPET_ADMIN_PASSWORD":"ASCII_password"}' --key-pair "key_pair_name" --
preferred-maintenance-window "ddd:hh:mm" --preferred-backup-window "ddd:hh:mm"
 --security-group-ids security_group_id1 security_group_id2 --service-role-arn
 "service_role_ARN" --subnet-ids subnet_ID
```

Nell'esempio seguente viene creato un master Puppet che utilizza un dominio personalizzato.

```
aws opsworks-cm create-server \
    --engine "Puppet" \
    --engine-model "Monolithic" \
    --engine-version "2019" \
    --server-name "puppet-02" \
    --instance-profile-arn "arn:aws:iam::1019881987024:instance-profile/aws-
opsworks-cm-ec2-role" \
    --instance-type "c4.large" \
    --engine-attributes '{"PUPPET_ADMIN_PASSWORD":"zZZzDj2DLYXSZFRv1d"}' \
    --custom-domain "my-puppet-master.my-corp.com" \
    --custom-certificate "-----BEGIN CERTIFICATE----- EXAMPLEqEXAMPLE== -----END
 CERTIFICATE-----" \
    --custom-private-key "-----BEGIN RSA PRIVATE KEY----- EXAMPLEqEXAMPLE= -----END
 RSA PRIVATE KEY-----" \
    --key-pair "amazon-test"
    --preferred-maintenance-window "Mon:08:00" \
    --preferred-backup-window "Sun:02:00" \
    --security-group-ids sg-b00000001 sg-b0000008 \
    --service-role-arn "arn:aws:iam::044726508045:role/service-role/aws-opsworks-
cm-service-role" \
    --subnet-ids subnet-383daa71
```

4. OpsWorks for Puppet Enterprise impiega circa 15 minuti per creare un nuovo server. Nell'output del comando `create-server`, copiare il valore dell'attributo `Endpoint`. Di seguito è riportato un esempio.

```
"Endpoint": "puppet-2019-exampleexample.opsworks-cm.us-east-1.amazonaws.com"
```

Non annullare l'output del comando `create-server` né chiudere la sessione shell, poiché l'output può contenere informazioni importanti che non vengono visualizzate nuovamente. Per ottenere le password e lo starter kit dai risultati `create-server`, passare alla fase successiva.

5. Se hai scelto di fare in modo che OpsWorks Puppet Enterprise generi una password per te, puoi estrarla in un formato utilizzabile dai `create-server` risultati utilizzando un processore JSON come jq. Dopo aver installato jq, è possibile eseguire i seguenti comandi per estrarre la password di amministratore Puppet e lo starter kit. Se non si fornisce la password nella Fase 3, assicurarsi di salvare la password di amministratore estratta in una posizione sicura, ma comoda.

```
#Get the Puppet password:
cat resp.json | jq -r '.Server.EngineAttributes[] | select(.Name ==
 "PUPPET_ADMIN_PASSWORD") | .Value'

#Get the Puppet Starter Kit:
cat resp.json | jq -r '.Server.EngineAttributes[] | select(.Name ==
 "PUPPET_STARTER_KIT") | .Value' | base64 -D > starterkit.zip
```

> **Note**
>
> Non è possibile rigenerare un nuovo starter kit Puppet master nella AWS Management Console. Quando crei un master Puppet utilizzando il AWS CLI, esegui il `jq` comando precedente per salvare lo starter kit con codifica base64 nei risultati come file ZIP. `create-server`

6. Facoltativamente, se non hai estratto lo starter kit dai risultati dei `create-server` comandi, puoi scaricare un nuovo starter kit dalla pagina Proprietà del server nella console di for Puppet Enterprise. OpsWorks
7. Se non si utilizza un dominio personalizzato, passare alla fase successiva. Se utilizzi un dominio personalizzato con il server, crea una voce CNAME nello strumento di gestione DNS dell'azienda per indirizzare il dominio personalizzato all'endpoint OpsWorks for Puppet Enterprise che hai copiato nel passaggio 4. Non è possibile raggiungere o eseguire l'accesso al server con un dominio personalizzato fino a quando non si completa questa fase.
8. Al termine del processo di creazione del server, vai su Configurare il Puppet Master utilizzando lo Starter Kit.

Vedi anche

- Crea un Puppet Enterprise Master utilizzando il AWS CLI
- Esegui il backup e il ripristino di un OpsWorks server Puppet Enterprise
- CreateServernel CM API Reference AWS OpsWorks
- create-server in Riferimento ai comandi AWS CLI

Lavorare con i tag sulle AWS OpsWorks for Puppet Enterprise risorse

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

I tag sono parole o frasi che fungono da metadati per l'identificazione e l'organizzazione delle risorse AWS. In OpsWorks Puppet Enterprise, una risorsa può avere fino a 50 tag applicati dall'utente. Ciascun tag è formato da una chiave e da un valore opzionale. Puoi applicare tag alle seguenti risorse in OpsWorks for Puppet Enterprise:

- OpsWorks per i server Puppet Enterprise
- Backup di OpsWorks per i server Puppet Enterprise

I tag sulle AWS risorse possono aiutarti a tenere traccia dei costi, controllare l'accesso alle risorse, raggruppare le risorse per automatizzare le attività o organizzare le risorse per scopo o fase del ciclo di vita. Per ulteriori informazioni sui vantaggi dei tag, consulta Strategie di tagging in AWS Answers e Utilizzo di tag per l'allocazione costi nella Guida utente di AWS Billing and Cost Management .

Per utilizzare i tag per controllare l'accesso ai server o ai OpsWorks backup di Puppet Enterprise, crea o modifica le istruzioni politiche in (IAM). AWS Identity and Access Management Per ulteriori

informazioni, consulta Controllo dell'accesso alle risorse AWS mediante i tag delle risorse nella Guida utente di AWS Identity and Access Management .

Quando applichi tag a un master OpsWorks for Puppet Enterprise, i tag vengono applicati anche ai backup del master, al bucket Amazon S3 che archivia i backup, all'istanza EC2 Amazon del master, ai segreti per il master archiviati AWS Secrets Manager in e all'indirizzo IP elastico utilizzato dal master. I tag non vengono propagati AWS CloudFormation nello stack utilizzato per creare il tuo master Puppet. AWS OpsWorks

Argomenti

- Come funzionano i tag in AWS OpsWorks for Puppet Enterprise
- Aggiungi e gestisci tag in OpsWorks Puppet Enterprise (console)
- Aggiungi e gestisci tag in OpsWorks Puppet Enterprise (CLI)
- Vedi anche

Come funzionano i tag in AWS OpsWorks for Puppet Enterprise

In questa versione, è possibile aggiungere e gestire i tag utilizzando l'API CM AWS OpsWorks o AWS Management Console. AWS OpsWorks CM tenta inoltre di aggiungere tag aggiunti a un server alle AWS risorse associate al server, tra cui l' EC2 istanza, i segreti in Secrets Manager, l'indirizzo IP elastico, il gruppo di sicurezza, il bucket S3 e i backup.

La tabella seguente fornisce una panoramica di come aggiungere e gestire i tag in OpsWorks Puppet Enterprise.

Azione	Cosa usare
Aggiungi tag a un nuovo server OpsWorks for Puppet Enterprise o a un backup che stai creando manualmente.	• Scegliere Crea server Puppet Enterprise e aggiungere tag nella pagina Configura impostazioni avanzate . • Scegliere Crea backup nella pagina Backup per un server esistente e aggiungere tag nella pagina Crea un backup del server Puppet Enterprise. • Aggiungere un parametro `Tags` ai comandi `CreateServer` o `CreateBackup` .

Azione	Cosa usare
Visualizza i tag su una risorsa.	• Nella pagina dei dettagli del server scegliere Tag nel riquadro di spostamento. • Nella pagina Backup del server selezionare un backup, quindi scegliere Modifica backup. • Esegui il comando `ListTagsForResource`.
Aggiungi tag a un server OpsWorks for Puppet Enterprise esistente o a un backup, indipendentemente dal fatto che il backup sia stato creato manualmente o automaticamente.	• Nella pagina dei dettagli del server scegliere Tag nel riquadro di spostamento, quindi scegliere Modifica. • Nella pagina Backup del server selezionare un backup, quindi scegliere Modifica backup. • Esegui il comando `TagResource`.
Per eliminare tag da una risorsa	• Nella pagina dei dettagli del server scegliere Tag nel riquadro di spostamento, quindi scegliere Modifica. Scegliere X accanto ai tag che si desidera eliminare. • Nella pagina Backup del server selezionare un backup, quindi scegliere Modifica backup. Scegliere X accanto ai tag che si desidera eliminare. • Esegui il comando `UntagResource`.

Le risposte `DescribeServers` e `DescribeBackups` non includono informazioni sui tag. Per visualizzare i tag, utilizzare l'API `ListTagsForResource`.

Aggiungi e gestisci tag in OpsWorks Puppet Enterprise (console)

Le procedure in questa sezione vengono eseguite in AWS Management Console.

Se si aggiungono tag, una chiave tag non può essere vuota. La chiave può avere un massimo di 127 caratteri e può contenere solo lettere, numeri o separatori Unicode oppure i seguenti caratteri speciali: `+ - = . _ : / @` i valori dei tag sono opzionali. È possibile aggiungere un tag che ha una

chiave, ma senza valori. Il valore può avere un massimo di 255 caratteri e può contenere solo lettere, numeri o separatori Unicode oppure i seguenti caratteri speciali: `+ - = . _ : / @`.

Argomenti

- Aggiungi tag a un nuovo server OpsWorks for Puppet Enterprise (Console)
- Aggiungere tag a un nuovo backup (console)
- Aggiungere o visualizzare tag su un server esistente (Console)
- Aggiungere o visualizzare tag su un backup esistente (Console)
- Eliminare tag da un server (Console)
- Eliminare tag da un backup (Console)

Aggiungi tag a un nuovo server OpsWorks for Puppet Enterprise (Console)

1. Assicurati di completare tutti i prerequisiti per creare un master OpsWorks for Puppet Enterprise.
2. Seguire i passaggi 1-8 in Crea un Puppet Enterprise Master utilizzando il AWS Management Console.
3. Dopo aver specificato le impostazioni di backup automatizzato, aggiungere i tag nell'area Tag della pagina Configura impostazioni avanzate . È possibile aggiungere un massimo di 50 tag. Una volta aggiunti i tag, scegliere Next (Avanti).
4. Passare al passaggio 11 di Crea un Puppet Enterprise Master utilizzando il AWS Management Console ed esaminare le impostazioni scelte per il nuovo server.

Aggiungere tag a un nuovo backup (console)

1. Nella home page OpsWorks di Puppet Enterprise, scegli un Puppet master esistente.
2. Nella pagina dei dettagli del server scegliere Backup nel riquadro di spostamento.
3. Nella pagina Backup scegliere Crea backup.
4. Aggiungi tag. Al termine dell'aggiunta delle regole, scegliere Crea.

Aggiungere o visualizzare tag su un server esistente (Console)

1. Nella home page OpsWorks di Puppet Enterprise, scegli un master Puppet esistente per aprirne la pagina dei dettagli.

2. Scegliere Tag nel riquadro di spostamento oppure, nella parte inferiore della pagina dei dettagli, scegliere Visualizza tutti i tag.
3. Nella pagina Tag scegliere Modifica.
4. Aggiungere o modificare tag sul server. Al termine, selezionare Save (Salva).

> **Note**
>
> Tieni presente che la modifica dei tag sul tuo Puppet master modifica anche i tag sulle risorse associate al server, come l' EC2 istanza, l'indirizzo IP elastico, il gruppo di sicurezza, il bucket S3 e i backup.

Aggiungere o visualizzare tag su un backup esistente (Console)

1. Nella home page OpsWorks di Puppet Enterprise, scegli un master Puppet esistente per aprirne la pagina dei dettagli.
2. Scegliere Backup nel riquadro di spostamento oppure, nell'area Backup recenti della pagina dei dettagli, scegliere Visualizza tutti i backup.
3. Nella pagina Backup scegliere un backup da gestire, quindi scegliere Modifica backup.
4. Aggiungere o modificare tag sul backup. Scegliere Aggiorna al termine.

Eliminare tag da un server (Console)

1. Nella home page OpsWorks di Puppet Enterprise, scegli un master Puppet esistente per aprirne la pagina dei dettagli.
2. Scegliere Tag nel riquadro di spostamento oppure, nella parte inferiore della pagina dei dettagli, scegliere Visualizza tutti i tag.
3. Nella pagina Tag scegliere Modifica.
4. Scegliere X accanto a un tag per eliminare il tag. Al termine, selezionare Save (Salva).

> **Note**
>
> Tieni presente che la modifica dei tag sul tuo Puppet master modifica anche i tag sulle risorse associate al server, come l' EC2 istanza, l'indirizzo IP elastico, il gruppo di sicurezza, il bucket S3 e i backup.

Eliminare tag da un backup (Console)

1. Nella home page OpsWorks di Puppet Enterprise, scegli un master Puppet esistente per aprirne la pagina dei dettagli.
2. Scegliere Backup nel riquadro di spostamento oppure, nell'area Backup recenti della pagina dei dettagli, scegliere Visualizza tutti i backup.
3. Nella pagina Backup scegliere un backup da gestire, quindi scegliere Modifica backup.
4. Scegliere X accanto a un tag per eliminare il tag. Scegliere Aggiorna al termine.

Aggiungi e gestisci tag in OpsWorks Puppet Enterprise (CLI)

Le procedure in questa sezione vengono eseguite in AWS CLI. Assicurati di utilizzare l'ultima versione di AWS CLI prima di iniziare a lavorare con i tag. Per ulteriori informazioni sull'installazione o l'aggiornamento di AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Se si aggiungono tag, una chiave tag non può essere vuota. La chiave può avere un massimo di 127 caratteri e può contenere solo lettere, numeri o separatori Unicode oppure i seguenti caratteri speciali: `+ - = . _ : / @` i valori dei tag sono opzionali. È possibile aggiungere un tag che ha una chiave, ma senza valori. Il valore può avere un massimo di 255 caratteri e può contenere solo lettere, numeri o separatori Unicode oppure i seguenti caratteri speciali: `+ - = . _ : / @`.

Argomenti

- Aggiungi tag a un nuovo server OpsWorks for Puppet Enterprise (CLI)
- Aggiungere tag a un nuovo backup (CLI)
- Aggiungere tag a server o backup esistenti (CLI)
- Elenca i tag delle risorse (CLI)
- Per eliminare tag da una risorsa (CLI)

Aggiungi tag a un nuovo server OpsWorks for Puppet Enterprise (CLI)

Puoi utilizzarlo AWS CLI per aggiungere tag quando crei un server OpsWorks for Puppet Enterprise. Questa procedura non descrive in modo completo come creare un server. Per informazioni dettagliate su come creare un server OpsWorks for Puppet Enterprise utilizzando il AWS CLI, consulta questa Crea un Puppet Enterprise Master utilizzando il AWS CLI guida. È possibile aggiungere fino a 50 tag a un server.

1. Assicurati di completare tutti i prerequisiti per creare un server OpsWorks for Puppet Enterprise.
2. Completa i passaggi 1-4 di Crea un Puppet Enterprise Master utilizzando il AWS CLI.
3. Per il passaggio 5, quando si esegue il comando `create-server`, aggiungere il parametro `--tags` al comando, come illustrato nell'esempio seguente.

   ```
   aws opsworks-cm create-server ... --tags Key=Key1,Value=Value1
    Key=Key2,Value=Value2
   ```

   Di seguito è riportato un esempio che mostra solo la parte dei tag del comando `create-server`.

   ```
   aws opsworks-cm create-server ... --tags Key=Stage,Value=Production
    Key=Department,Value=Marketing
   ```
4. Completare le fasi restanti in Crea un Puppet Enterprise Master utilizzando il AWS CLI. Per verificare che i tag siano stati aggiunti al nuovo server, attenersi alla procedura descritta in Elenca i tag delle risorse (CLI) in questo argomento.

Aggiungere tag a un nuovo backup (CLI)

Puoi utilizzarlo AWS CLI per aggiungere tag quando crei un nuovo backup manuale di un server OpsWorks for Puppet Enterprise. Questa procedura non descrive in modo completo come creare un backup manuale. Per informazioni dettagliate su come creare un backup manuale, vedi «Per eseguire un backup manuale nel AWS CLI" inEsegui il backup di un OpsWorks Puppet Enterprise Server. È possibile aggiungere fino a 50 tag a un backup. Se un server dispone di tag, ai nuovi backup vengono automaticamente applicati i tag del server.

Per impostazione predefinita, quando crei un nuovo server OpsWorks for Puppet Enterprise, i backup automatici sono abilitati. È possibile aggiungere tag a un backup automatico eseguendo il comando `tag-resource` descritto in Aggiungere tag a server o backup esistenti (CLI) in questo argomento.

- Per aggiungere tag a un backup manuale durante la creazione del backup, eseguire il comando seguente. Viene visualizzata solo la parte dei tag del comando. Per un esempio del comando `create-backup` completo, vedere «Per eseguire un backup manuale in AWS CLI" in Esegui il backup di un OpsWorks Puppet Enterprise Server.

  ```
  aws opsworks-cm create-backup ... --tags Key=Key1,Value=Value1
   Key=Key2,Value=Value2
  ```

L'esempio seguente mostra solo la parte tag del comando `create-backup`.

```
aws opsworks-cm create-backup ... --tags Key=Stage,Value=Production
 Key=Department,Value=Marketing
```

Aggiungere tag a server o backup esistenti (CLI)

Puoi eseguire il `tag-resource` comando per aggiungere tag ai server o ai backup esistenti OpsWorks di Puppet Enterprise (indipendentemente dal fatto che i backup siano stati creati automaticamente o manualmente). Specificare il numero di risorsa Amazon (ARN) di una risorsa di destinazione per aggiungervi dei tag.

1. Per ottenere l'ARN della risorsa a cui si desidera applicare i tag:

   - Per un server, eseguire `describe-servers --server-name` *`server_name`*. I risultati del comando mostrano il server ARN.
   - Per un backup, eseguire `describe-backups --backup-id` *`backup_ID`*. I risultati del comando mostrano l'ARN di backup. Puoi anche eseguire l'operazione `describe-backups --server-name` *`server_name`* per mostrare informazioni su tutti i backup per uno specifico OpsWorks server Puppet Enterprise.

   L'esempio seguente mostra solo `ServerArn` nei risultati di un comando `describe-servers --server-name opsworks-cm-test`. Il valore `ServerArn` viene aggiunto a un comando `tag-resource` per aggiungere tag al server.

   ```
   {
       "Servers": [
           {
               ...
               "ServerArn": "arn:aws:opsworks-cm:us-west-2:123456789012:server/
   opsworks-cm-test/EXAMPLEd-66b0-4196-8274-d1a2bEXAMPLE"
           }
       ]
   }
   ```

2. Eseguire il comando `tag-resource` con l'ARN restituito nel passaggio 1.

```
aws opsworks-cm tag-resource --resource-arn "server_or_backup_ARN" --tags
 Key=Key1,Value=Value1 Key=Key2,Value=Value2
```

Di seguito è riportato un esempio.

```
aws opsworks-cm tag-resource --resource-arn "arn:aws:opsworks-cm:us-
west-2:123456789012:server/opsworks-cm-test/EXAMPLEd-66b0-4196-8274-d1a2bEXAMPLE"
 --tags Key=Stage,Value=Production Key=Department,Value=Marketing
```

3. Per verificare che i tag siano stati aggiunti correttamente, passare alla procedura successiva Elenca i tag delle risorse (CLI).

Elenca i tag delle risorse (CLI)

Puoi eseguire il `list-tags-for-resource` comando per mostrare i tag a cui sono allegati i server o i backup OpsWorks di Puppet Enterprise. Specificare l'ARN di una risorsa di destinazione per visualizzarne i tag.

1. Per ottenere l'ARN della risorsa per la quale si desidera elencare i tag:
   - Per un server, eseguire `describe-servers --server-name` *`server_name`*. I risultati del comando mostrano il server ARN.
   - Per un backup, eseguire `describe-backups --backup-id` *`backup_ID`*. I risultati del comando mostrano l'ARN di backup. Puoi anche eseguire l'operazione `describe-backups --server-name` *`server_name`* per mostrare informazioni su tutti i backup per uno specifico server Puppet OpsWorks Enterprise.
2. Eseguire il comando `list-tags-for-resource` con l'ARN restituito nel passaggio 1.

```
aws opsworks-cm list-tags-for-resource --resource-arn "server_or_backup_ARN"
```

Di seguito è riportato un esempio.

```
aws opsworks-cm tag-resource --resource-arn "arn:aws:opsworks-cm:us-
west-2:123456789012:server/opsworks-cm-test/EXAMPLEd-66b0-4196-8274-d1a2bEXAMPLE"
```

Se sono presenti tag sulla risorsa, il comando restituisce risultati come i seguenti.

```
{
    "Tags": [
        {
            "Key": "Stage",
            "Value": "Production"
        },
        {
            "Key": "Department",
            "Value": "Marketing"
        }
    ]
}
```

Per eliminare tag da una risorsa (CLI)

Puoi eseguire il `untag-resource` comando per eliminare i tag dai server o dai backup OpsWorks di Puppet Enterprise. Se la risorsa viene eliminata, anche i tag sulla risorsa vengono eliminati. Specificare l'Amazon Resource Number (ARN) di una risorsa di destinazione per rimuovere i tag da essa.

1. Per ottenere l'ARN della risorsa da cui si desidera rimuovere i tag:
   - Per un server, eseguire `describe-servers --server-name` *`server_name`*. I risultati del comando mostrano il server ARN.
   - Per un backup, eseguire `describe-backups --backup-id` *`backup_ID`*. I risultati del comando mostrano l'ARN di backup. Puoi anche eseguire l'operazione `describe-backups --server-name` *`server_name`* per mostrare informazioni su tutti i backup per uno specifico server Puppet OpsWorks Enterprise.
2. Eseguire il comando `untag-resource` con l'ARN restituito nel passaggio 1. Specificare solo i tag che si desidera eliminare.

   ```
   aws opsworks-cm untag-resource --resource-arn "server_or_backup_ARN" --tags
    Key=Key1,Value=Value1 Key=Key2,Value=Value2
   ```

   In questo esempio, il comando `untag-resource` rimuove solo il tag con una chiave `Stage` e un valore pari a `Production`.

```
aws opsworks-cm untag-resource --resource-arn "arn:aws:opsworks-cm:us-
west-2:123456789012:server/opsworks-cm-test/EXAMPLEd-66b0-4196-8274-d1a2bEXAMPLE"
 --tags Key=Stage,Value=Production
```

3. Per verificare che i tag siano stati eliminati correttamente, attenersi alla procedura descritta in Elenca i tag delle risorse (CLI) in questo argomento.

Vedi anche

- Crea un Puppet Enterprise Master utilizzando il AWS CLI
- Esegui il backup di un OpsWorks Puppet Enterprise Server
- AWS Tagging Strategies (Strategie di tagging AWS)
- Controllo dell'accesso alle AWS risorse utilizzando i tag delle risorse nella Guida per l'AWS Identity and Access Management utente
- Uso dei tag per l'allocazione dei costi nella Guida per l'utente di AWS Billing and Cost Management .
- CreateBackup in Documentazione di riferimento dell'API CM di AWS OpsWorks
- CreateServer in Documentazione di riferimento dell'API CM di AWS OpsWorks
- TagResource in Documentazione di riferimento dell'API CM di AWS OpsWorks
- ListTagsForResource in Documentazione di riferimento dell'API CM di AWS OpsWorks
- UntagResource in Documentazione di riferimento dell'API CM di AWS OpsWorks

Esegui il backup e il ripristino di un OpsWorks server Puppet Enterprise

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione descrive come eseguire il backup e ripristino di un server OpsWorks per Puppet Enterprise.

Argomenti

- Esegui il backup di un OpsWorks Puppet Enterprise Server
- Ripristina un server OpsWorks for Puppet Enterprise da un backup

Esegui il backup di un OpsWorks Puppet Enterprise Server

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi definire un backup ricorrente giornaliero o settimanale OpsWorks per il server Puppet Enterprise e fare in modo che il servizio archivi i backup in Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) per tuo conto. In alternativa, puoi effettuare backup manuali on demand.

Poiché i backup vengono archiviati in Amazon S3, comportano costi aggiuntivi. È possibile definire un periodo di retention dei backup fino a 30 generazioni. Puoi inviare una richiesta di assistenza per modificare tale limite utilizzando i canali di supporto. AWS I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Puoi aggiungere tag ai backup di un master OpsWorks for Puppet Enterprise. Se hai aggiunto tag a un master di OpsWorks for Puppet Enterprise, i backup automatici del master Puppet ereditano quei tag. Per ulteriori informazioni su come aggiungere e gestire i tag nei backup, vedere Lavorare con i tag sulle AWS OpsWorks for Puppet Enterprise risorse in questa guida.

Argomenti

- Backup automatici
- Backup manuali

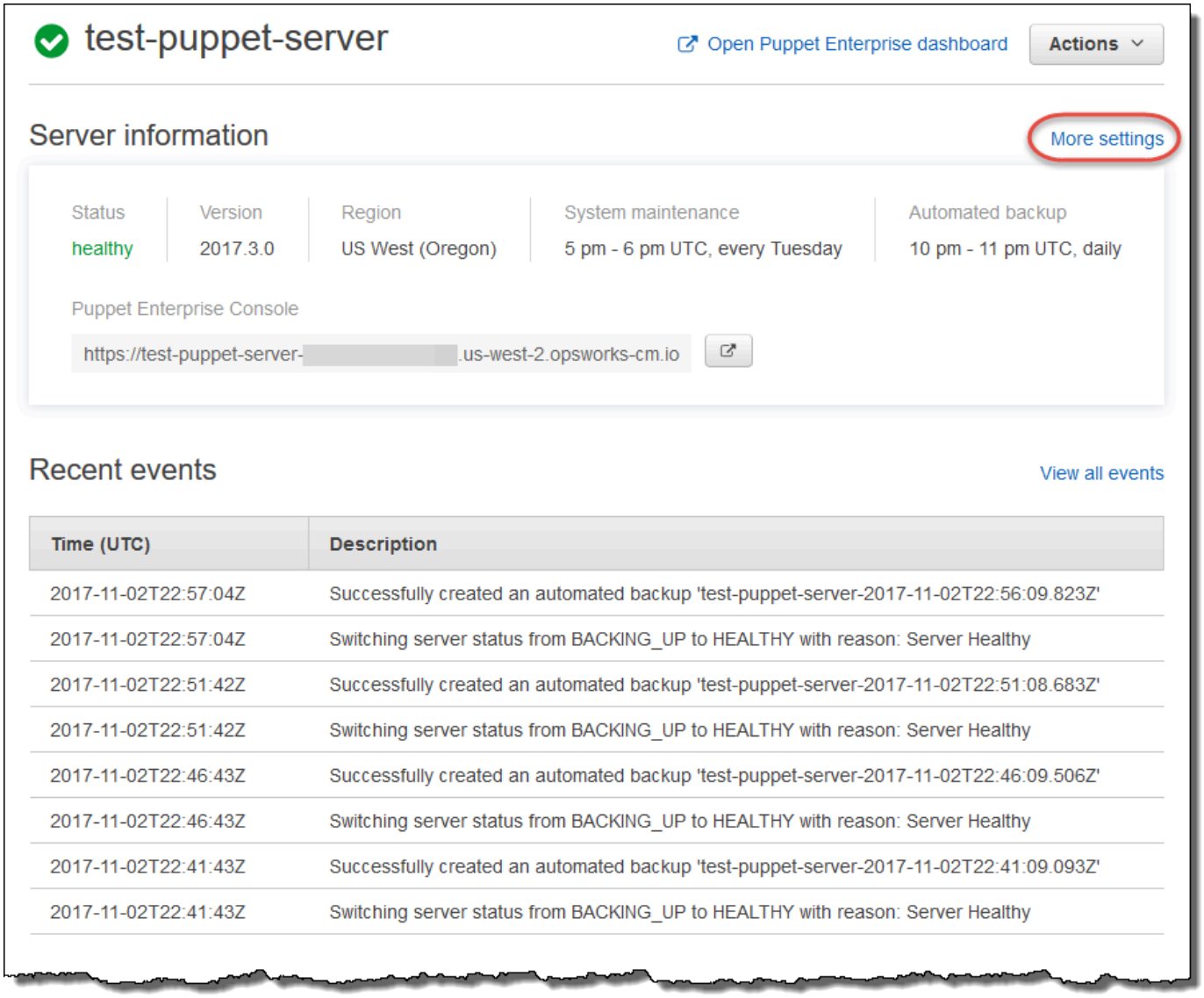
- Eliminare i backup

Backup automatici

Quando configuri il tuo server OpsWorks for Puppet Enterprise, scegli tra backup automatici o manuali. OpsWorks for Puppet Enterprise avvia i backup automatici nell'ora e nel giorno scelti nella sezione Backup automatico della pagina Configura impostazioni avanzate della procedura guidata di configurazione. Dopo che il server è online, puoi modificare le impostazioni di backup eseguendo le seguenti fasi nella pagina delle proprietà del server.

Per modificare le impostazioni del backup automatico

1. Nella pagina delle proprietà del server, scegliere More settings (Altre impostazioni).



2. Per disattivare i backup automatici, scegliere No per l'opzione Enable automated backups (Abilita backup automatici). Salvare le modifiche. Non è necessario procedere con il passaggio successivo.
3. Nella sezione Automated Backup (Backup automatico) modificare la frequenza, l'ora di inizio o le generazioni da conservare. Salvare le modifiche.

Backup manuali

Puoi avviare un backup manuale in qualsiasi momento in o eseguendo AWS Management Console il comando create-backup. AWS CLI I backup manuali non sono inclusi nel massimo 30 generazioni di backup automatizzati archiviati. Vengono archiviati al massimo 10 backup manuali che devono essere eliminati manualmente da Amazon S3.

Per eseguire un backup manuale in AWS Management Console

1. Nella pagina Puppet Enterprise servers (Server Puppet Enterprise), scegliere il server di cui si desidera eseguire il backup.
2. Nella pagina delle proprietà del server, nel riquadro di navigazione a sinistra, scegliere Backups (Backup).
3. Scegliere Create backup (Crea backup).
4. Il backup manuale finisce quando viene visualizzato un segno di spunta verde nella colonna Status (Stato) del backup.

Per eseguire un backup manuale in AWS CLI

Puoi aggiungere tag quando crei un nuovo backup manuale di un server OpsWorks per Puppet Enterprise. Per ulteriori informazioni su come aggiungere tag quando si crea un backup manuale, vedere Aggiungere tag a un nuovo backup (CLI).

- Per avviare un backup manuale, esegui il AWS CLI comando riportato di seguito.

```
aws opsworks-cm --region region name create-backup --server-name "Puppet server
 name" --description "optional descriptive string"
```

Eliminare i backup

L'eliminazione di un backup comporta la sua eliminazione definitiva dal bucket S3 in cui vengono archiviati i backup.

Per eliminare un backup in AWS Management Console

1. Nella pagina Puppet Enterprise servers (Server Puppet Enterprise), scegliere il server di cui si desidera eseguire il backup.
2. Nella pagina delle proprietà del server, nel riquadro di navigazione a sinistra, scegliere Backups (Backup).
3. Scegliere il backup da eliminare, quindi scegliere Delete backup (Elimina backup). Puoi selezionare un solo backup alla volta.
4. Quando viene richiesto di confermare l'eliminazione, selezionare la casella di controllo Delete the backup, which is stored in an S3 bucket (Elimina il backup archiviato in un bucket S3) e scegliere Yes, Delete (Sì, elimina).

Per eliminare un backup in AWS CLI

- Per eliminare un backup, esegui il AWS CLI comando seguente, sostituendo il valore di `--backup-id` con l'ID del backup che desideri eliminare. IDs I backup sono nel formato*`ServerName-yyyyMMddHHmmssSSS`*. Ad esempio, **`puppet-server-20171218132604388`**.

  ```
  aws opsworks-cm --region region name delete-backup --backup-id ServerName-yyyyMMddHHmmssSSS
  ```

Ripristina un server OpsWorks for Puppet Enterprise da un backup

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver esaminato i backup disponibili, puoi scegliere facilmente un momento temporale a partire dal quale ripristinare il tuo server OpsWorks for Puppet Enterprise. I backup del server contengono dati persistenti relativi al software di gestione della configurazione, ad esempio moduli, classi, associazioni di nodi, informazioni sui database (compresi rapporti, dati e così via). L'esecuzione di un ripristino sul posto di un server (ovvero il ripristino del server OpsWorks for Puppet Enterprise esistente su una nuova EC2 istanza) registra nuovamente i nodi che erano registrati al momento del backup utilizzato per ripristinare il server e trasferisce il traffico alla nuova istanza se il ripristino ha esito positivo e lo stato del server ripristinato OpsWorks per Puppet Enterprise è lo stesso`Healthy`. Il ripristino su un server Puppet Enterprise appena OpsWorks creato non mantiene le connessioni dei nodi. Il ripristino di un server non aggiorna la versione del software Puppet, ma applica le stesse versioni Puppet e gli stessi dati di gestione della configurazione disponibili nel backup scelto.

Il ripristino di un server richiede in genere più tempo rispetto alla creazione di un nuovo server; il tempo dipende dalla dimensione del backup scelto. Una volta completato il ripristino, la vecchia EC2 istanza rimane nello `Stopped` stato `Running` or, ma solo temporaneamente. Alla fine viene terminata.

In questa versione, puoi usare AWS CLI per ripristinare un Puppet master in Puppet OpsWorks Enterprise.

> **Note**
>
> Puoi eseguire inoltre il comando restore-server per modificare il tipo di istanza corrente o per ripristinare o configurare la chiave SSH qualora venga persa o compromessa.

Per ripristinare un server da un backup

1. In AWS CLI, esegui il comando seguente per restituire un elenco di backup disponibili e relativi. IDs Annotare l'ID del backup che si desidera utilizzare. IDs I backup sono nel formato*myServerName-yyyyMMddHHmmssSSS*.

   ```
   aws opsworks-cm --region region name describe-backups
   ```

2. Esegui il comando seguente.

   ```
   aws opsworks-cm --region region name restore-server --backup-id "myServerName-
   yyyyMMddHHmmssSSS" --instance-type "Type of instance" --key-pair "name of your EC2
    key pair" --server-name "name of Puppet master"
   ```

   Di seguito è riportato un esempio.

   ```
   aws opsworks-cm --region us-west-2 restore-server --backup-id
    "MyPuppetServer-20161120122143125" --server-name "MyPuppetServer"
   ```

3. Attendi il completamento del ripristino.

Manutenzione del sistema OpsWorks per Puppet Enterprise

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile.

Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

La manutenzione obbligatoria del sistema garantisce che le ultime versioni AWS testate di Puppet Server, inclusi gli aggiornamenti di sicurezza, siano sempre in esecuzione su un server OpsWorks for Puppet Enterprise. La manutenzione del sistema è richiesta almeno una volta alla settimana. Utilizzando AWS CLI, è possibile configurare la manutenzione automatica giornaliera, se lo si desidera. È inoltre possibile utilizzare il AWS CLI per eseguire la manutenzione del sistema su richiesta, oltre alla manutenzione programmata del sistema.

Quando diventano disponibili nuove versioni del software Puppet, la manutenzione del sistema è progettata per aggiornare automaticamente la versione di Puppet Server sul server, non appena supera i test AWS. AWS esegue test approfonditi per verificare che gli upgrade di Puppet siano pronti per la produzione e non interferiscano con gli ambienti dei clienti esistenti, quindi possono verificarsi ritardi tra le versioni del software Puppet e la loro disponibilità per l'applicazione ai server esistenti di Puppet Enterprise. OpsWorks Per aggiornare le versioni disponibili del software Puppet on demand, consulta Avvio della manutenzione del sistema on demand in questo argomento.

La manutenzione del sistema avvia una nuova istanza da un backup eseguito come parte del processo di manutenzione, il che aiuta a ridurre il rischio di EC2 istanze Amazon danneggiate o danneggiate sottoposte a manutenzione periodica.

Important

La manutenzione del sistema elimina tutti i file o le configurazioni personalizzate che hai aggiunto al server for Puppet Enterprise. OpsWorks Per ulteriori informazioni su come riparare la configurazione o ovviare alla perdita di file, consulta Ripristino di configurazioni e file personalizzati dopo la manutenzione in questo argomento.

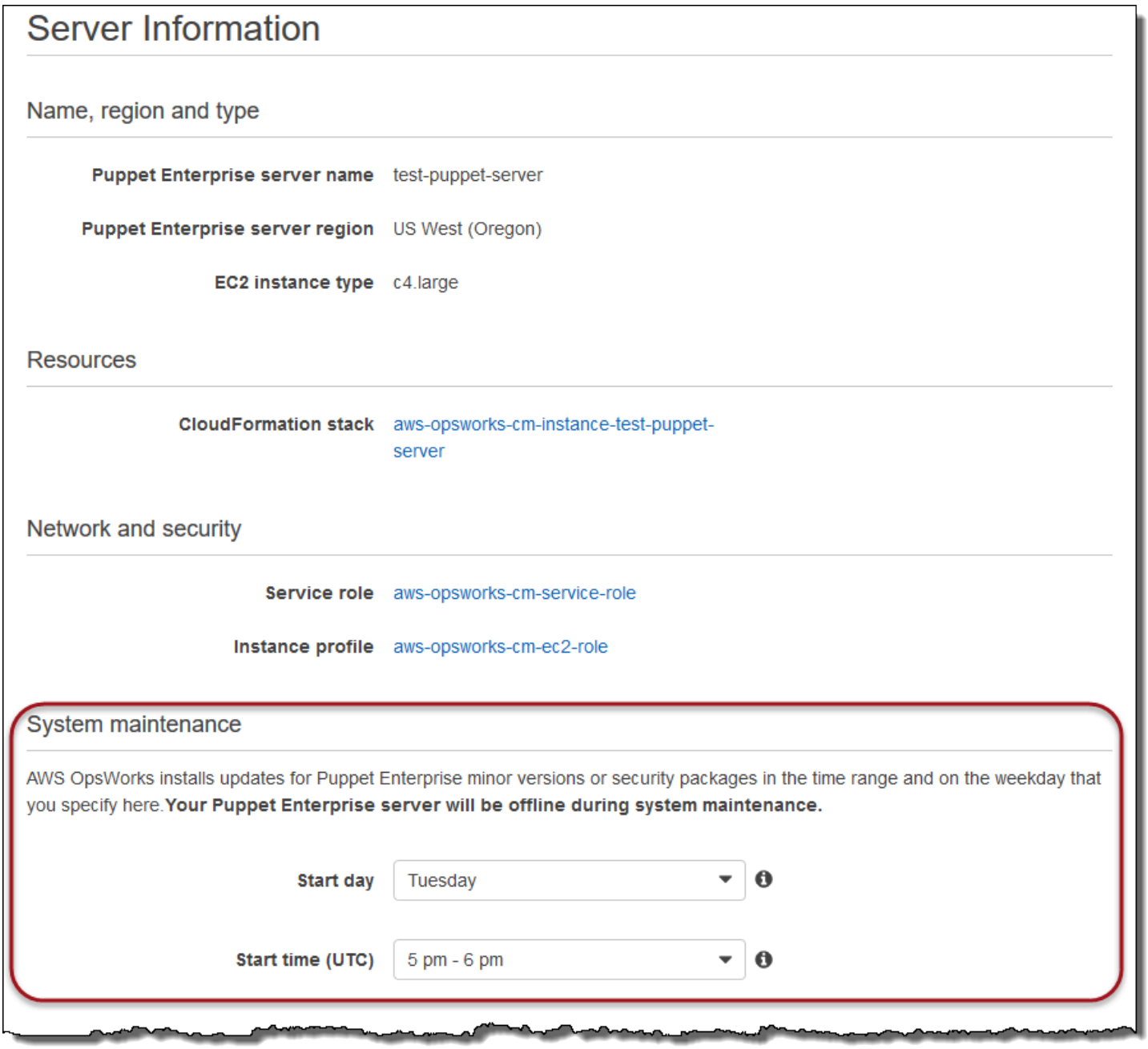
Argomenti

- Configurazione della manutenzione del sistema
- Avvio della manutenzione del sistema on demand
- Ripristino di configurazioni e file personalizzati dopo la manutenzione

Configurazione della manutenzione del sistema

Quando crei un nuovo server OpsWorks for Puppet Enterprise, puoi configurare un giorno della settimana e un'ora, in Coordinated Universal Time (UTC), per l'avvio della manutenzione del sistema. La manutenzione viene avviata all'ora specificata. In quanto ci si deve aspettare che il server sia offline durante il mantenimento del sistema, scegliere un orario di utilizzo del server basso durante l'orario di ufficio regolare. Durante la manutenzione, lo stato del server è `UNDER_MAINTENANCE`.

Puoi anche modificare le impostazioni di manutenzione del sistema su un server OpsWorks di Puppet Enterprise esistente, modificando le impostazioni nell'area Manutenzione del sistema della pagina Impostazioni del tuo server, come mostrato nella schermata seguente.



Nella sezione System maintenance (Manutenzione sistema), impostare il giorno e l'ora quando si desidera che il mantenimento del sistema inizi.

Configurazione della manutenzione del sistema utilizzando AWS CLI

Puoi anche configurare l'ora di avvio automatico per la manutenzione del sistema tramite AWS CLI. AWS CLI Consente di configurare la manutenzione automatica giornaliera, se lo si desidera, omettendo il prefisso di tre caratteri del giorno della settimana.

In un comando `create-server` aggiungi il parametro `--preferred-maintenance-window` dopo aver specificato i requisiti per la creazione dell'istanza del server (ad esempio tipo di istanza, ARN del profilo dell'istanza e ARN del ruolo di servizio). Nell'esempio `create-server` seguente, `--preferred-maintenance-window` è impostato su `Mon:08:00`, quindi la manutenzione è stata impostata per essere avviata ogni lunedì mattina alle 8:00 UTC.

```
aws opsworks-cm create-server --engine "Puppet" --engine-model "Monolithic"
 --engine-version "2017" --server-name "puppet-06" --instance-profile-arn
 "arn:aws:iam::1119001987000:instance-profile/aws-opsworks-cm-ec2-role"
 --instance-type "c4.large" --key-pair "amazon-test" --service-role-arn
 "arn:aws:iam::044726508045:role/aws-opsworks-cm-service-role" --preferred-maintenance-
window "Mon:08:00"
```

In un comando `update-server` puoi aggiornare solo il valore `--preferred-maintenance-window`, se lo desideri. Nell'esempio seguente la finestra di manutenzione è impostata su venerdì sera alle 18:15 UTC.

```
aws opsworks-cm update-server --server-name "puppet-06" --preferred-maintenance-window
 "Fri:18:15"
```

Per modificare l'ora di inizio della finestra di manutenzione in modo che sia impostata sulle 18:15 UTC tutti i giorni, ometti il prefisso di tre caratteri relativo al giorno feriale, come illustrato nell'esempio seguente.

```
aws opsworks-cm update-server --server-name "puppet-06" --preferred-maintenance-window
 "18:15"
```

Per ulteriori informazioni sull'impostazione della finestra preferita di manutenzione del sistema utilizzando AWS CLI, consulta create-server e update-server.

Avvio della manutenzione del sistema on demand

Per avviare la manutenzione del sistema su richiesta, al di fuori della manutenzione automatica settimanale o giornaliera configurata, esegui il comando seguente. AWS CLI Non è possibile avviare la manutenzione on-demand nella AWS Management Console.

```
aws opsworks-cm start-maintenance --server-name server_name
```

Per ulteriori informazioni su questo comando, consulta start-maintenance.

Ripristino di configurazioni e file personalizzati dopo la manutenzione

La manutenzione del sistema può eliminare o modificare file o configurazioni personalizzati che hai aggiunto al tuo server OpsWorks for Puppet Enterprise.

Se, dopo un'esecuzione di manutenzione, al tuo Puppet master mancano file o impostazioni che hai aggiunto utilizzando `RunCommand` o SSH, puoi utilizzare un'Amazon Machine Image (AMI) per avviare una nuova istanza Amazon EC2 . AMIs sono disponibili basate sulla configurazione di pre-manutenzione di un server.

La nuova istanza si trova nello stesso stato del master Puppet prima della manutenzione e dovrebbe includere le impostazioni e i file mancanti.

> **Important**
>
> Non è possibile utilizzare la nuova istanza per ripristinare il server. L'istanza non può essere eseguita come master Puppet. Puoi utilizzare l'istanza solo per ripristinare le impostazioni di configurazione e i file.

Per avviare un' EC2 istanza da un'AMI, nella EC2 console Amazon, apri la procedura guidata di avvio, scegli Mio AMIs, quindi scegli l'AMI con il nome del tuo server. Segui i passaggi della EC2 procedura guidata di Amazon come faresti per il lancio di qualsiasi altra istanza.

Aggiungere nodi automaticamente in OpsWorks Puppet Enterprise

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo argomento descrive come aggiungere automaticamente nodi Amazon Elastic Compute Cloud (Amazon EC2) al server OpsWorks for Puppet Enterprise. In Aggiungere nodi che il Puppet Master può gestire è stato illustrato come utilizzare il comando `associate-node` per aggiungere un nodo

alla volta al server Puppet Enterprise. Il codice in questo argomento mostra come aggiungere nodi automaticamente tramite il metodo automatico. Il metodo consigliato per l'associazione automatica (o automatica) di nuovi nodi consiste nel configurare i dati EC2 utente di Amazon. Per impostazione predefinita, un server OpsWorks for Puppet Enterprise è già `puppet-agent`disponibile per i sistemi operativi dei nodi Ubuntu, Amazon Linux e RHEL.

Per informazioni su come dissociare un nodo, consulta Dissocia un nodo da un server OpsWorks for Puppet Enterprise questa guida e la documentazione relativa `disassociate-node`all'API Puppet OpsWorks Enterprise.

Passaggio 1: crea un ruolo IAM da utilizzare come profilo di istanza

Crea un ruolo AWS Identity and Access Management (IAM) da utilizzare come profilo di EC2 istanza e collega la seguente policy al ruolo IAM. Questa policy consente all'`opsworks-cm`API di comunicare con l' EC2 istanza durante la registrazione del nodo. Per ulteriori informazioni sui profili di istanza, consulta Using Instance Profiles nella EC2 documentazione di Amazon. Per informazioni su come creare un ruolo IAM, consulta Creazione di un ruolo IAM nella console nella EC2 documentazione di Amazon.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "opsworks-cm:AssociateNode",
                "opsworks-cm:DescribeNodeAssociationStatus",
                "opsworks-cm:DescribeServers",
                "ec2:DescribeTags"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Fase 2. Creazione delle istanze tramite uno script di associazione automatica

Per creare EC2 istanze, puoi copiare lo script dei dati utente incluso nello Starter Kit nella `userdata` sezione delle istruzioni dell' EC2istanza, nelle configurazioni di avvio del gruppo Amazon EC2

Auto Scaling o in un modello. AWS CloudFormation Lo script è supportato solo per EC2 le istanze che eseguono i sistemi operativi Ubuntu e Amazon Linux. Per ulteriori informazioni sull'aggiunta di script ai dati utente, consulta Running Commands on Your Linux Instance at Launch nella EC2 documentazione di Amazon. Il modo più semplice per creare un nuovo nodo è utilizzare la procedura guidata di avvio delle EC2 istanze Amazon. Questa procedura dettagliata utilizza la configurazione di esempio del modulo del server Web Apache descritta in. Guida introduttiva a OpsWorks for Puppet Enterprise

1. Lo script dei dati utente nello Starter Kit esegue il `associate-node`comando `opsworks-cm` API per associare un nuovo nodo al tuo Puppet master. In questa versione, installa automaticamente anche la versione corrente di AWS CLI on the node, nel caso in cui non sia già in esecuzione la maggior parte della versione. up-to-date Salvare questo script in una posizione comoda con il nome `userdata.sh`.

   Per impostazione predefinita, il nome del nuovo nodo registrato corrisponde all'ID dell'istanza.
2. Segui la procedura riportata in Launching an Instance nella EC2 documentazione, con le modifiche apportate qui. Nella procedura guidata di avvio dell' EC2 istanza, scegli un'AMI Amazon Linux.
3. Nella pagina Configure Instance Details (Configura dettagli dell'istanza), selezionare myPuppetinstanceprofile, ovvero il ruolo creato in Passaggio 1: crea un ruolo IAM da utilizzare come profilo di istanza come ruolo IAM.
4. Nell'area Advanced Details (Dettagli avanzati), cariare lo script `userdata.sh` creato nella fase 1.
5. Non è necessario apportare alcuna modifica nella pagina Add Storage (Aggiungi storage). Vai su Add Tags (Aggiungi tag).

   Applicando i tag alla tua EC2 istanza, puoi personalizzare il comportamento di`userdata.sh`. In questo esempio, applica il ruolo `apache_webserver` al nodo mediante l'aggiunta del seguente tag: **`pp_role`**, con il valore **`apache_webserver`**.

   L'impostazione del valore `pp_role` sul nodo imposta i valori archiviati in modo permanente nel certificato dell'agente del nodo. In questo modo si abilita la classificazione affidabile del nodo. Per ulteriori informazioni, consulta l'argomento relativo alle richieste di estensione (dati di certificato permanenti) nella documentazione della piattaforma Puppet.
6. Nella pagina Configura gruppo di sicurezza, scegli Aggiungi regola, quindi scegli il tipo HTTP per aprire la porta 8080 per il server web Apache in questo esempio.

7. Scegli Review and Launch (Analizza e lancia), quindi scegli Launch (Lancia). All'avvio del nuovo nodo, applica la configurazione Apache del modulo di esempio in cui è stato configurato. Configurare l'esempio Apache dello Starter Kit
8. Quando apri la pagina web collegata al DNS pubblico del tuo nuovo nodo, dovresti vedere un sito web ospitato dal tuo server web Apache gestito da Puppet.

Dissocia un nodo da un server OpsWorks for Puppet Enterprise

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione descrive come dissociare o rimuovere un nodo gestito dalla gestione da parte di un server OpsWorks for Puppet Enterprise. Questa operazione viene eseguita sulla riga di comando o nella console di Puppet Enterprise; non è possibile dissociare i nodi nella console di gestione di Puppet Enterprise. OpsWorks Attualmente, l'API OpsWorks for Puppet Enterprise non consente la rimozione in batch di più nodi. Il comando descritto in questa sezione consente di eliminare l'associazione di un nodo alla volta.

Consigliamo di eliminare l'associazione dei nodi da un master Puppet prima di eliminare il server. In questo modo i nodi continueranno a funzionare senza tentare di riconnettersi al server. Per fare ciò, esegui il `disassociate-node` AWS CLI comando. Per rimuovere completamente un nodo da PE, devi eliminare l'associazione del nodo e revocarne il certificato, in modo che il nodo non cerchi continuamente di eseguire il check-in nel master Puppet. Devi inoltre disinstallare `puppet-agent` dai nodi quando non vuoi più gestirli utilizzando il master Puppet.

Per disassociare i nodi

1. In AWS CLI, esegui il comando seguente per dissociare i nodi. *`Node_name`*è il nome del nodo da cui desideri dissociare; per EC2 le istanze Amazon, questo è l'ID dell'istanza. *`Server_name`*è il nome del Puppet master da cui desideri dissociare il nodo. Entrambi i parametri sono obbligatori.

Il parametro `--region` non è obbligatorio a meno che non si desidera dissociare un nodo da un master Puppet che non si trova nella regione predefinita.

```
aws opsworks-cm --region Region_name disassociate-node --node-name Node_name --
server-name Server_name
```

Il comando seguente è un esempio.

```
aws opsworks-cm --region us-west-2 disassociate-node --node-name
 i-0010zzz00d66zzz90 --server-name opsworkstest
```

2. Attendere un messaggio di risposta in cui viene indicato che la disassociazione è terminata.

Per ulteriori informazioni su come eliminare un server OpsWorks for Puppet Enterprise, consulta. Eliminare un file OpsWorks per Puppet Enterprise Server

Vedi anche

- Remove nodes (Rimozione di nodi) nella documentazione di Puppet Enterprise

Eliminare un file OpsWorks per Puppet Enterprise Server

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione descrive come eliminare un server OpsWorks for Puppet Enterprise. L'eliminazione di un server comporta anche l'eliminazione dei relativi eventi, log e moduli, memorizzati nel server. Vengono eliminate anche le risorse di supporto (istanza Amazon Elastic Compute Cloud, volume Amazon Elastic Block Store, ecc.), insieme a tutti i backup automatici.

Sebbene l'eliminazione di un server non comporti l'eliminazione dei nodi, questi ultimi non verranno più gestiti dal server eliminato e tenteranno continuamente di riconnettersi. Per questo motivo, ti

consigliamo di disassociare i nodi gestiti prima di eliminare un master Puppet. In questa versione, puoi dissociare i nodi eseguendo un comando. AWS CLI

Passaggio 1: Disassociare i nodi gestiti

Dissocia i nodi dal master Puppet prima di eliminare il server. In questo modo i nodi continueranno a funzionare senza tentare di riconnettersi al server. A tale scopo, esegui il `disassociate-node` AWS CLI comando.

Per disassociare i nodi

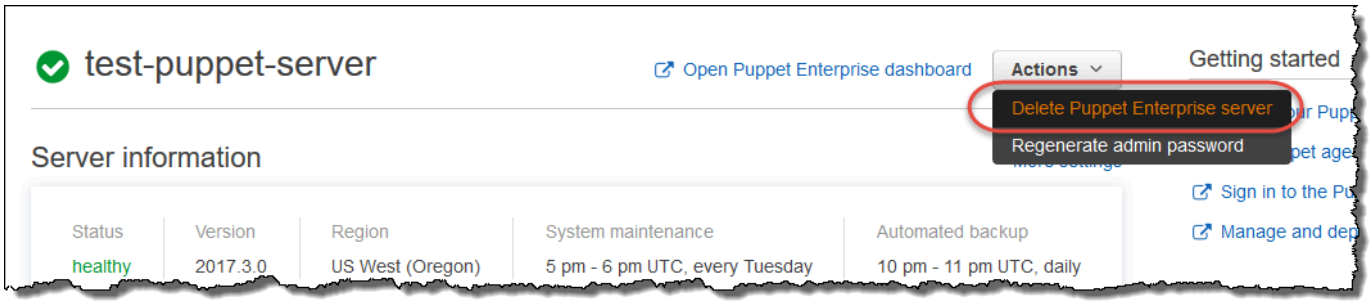
1. In AWS CLI, esegui il comando seguente per dissociare i nodi. *`Server_name`*è il nome del Puppet master da cui desideri dissociare il nodo. Il valore di `--node-name` può essere un ID istanza.

   ```
   aws opsworks-cm --region Region_name disassociate-node --node-name Node_name --
   server-name Server_name
   ```

2. Attendere un messaggio di risposta in cui viene indicato che la disassociazione è terminata.

Passaggio 2: Eliminare il server

1. Nel riquadro del server nel pannello di controllo, espandere il menu Actions (Azioni).



2. Scegliere Delete Puppet Enterprise server (Elimina Puppet Enterprise server).
3. Quando viene richiesto di confermare l'eliminazione, selezionare la casella di controllo per eliminare ruoli e risorse associati, quindi scegliere Yes, Delete (Sì, elimina).

Vedi anche

- Dissocia un nodo da un server OpsWorks for Puppet Enterprise

Come migrare un server OpsWorks for Puppet Enterprise su Amazon Elastic Compute Cloud (Amazon) EC2

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le istruzioni seguenti descrivono come migrare i server Puppet Enterprise esistenti su Amazon EC2, nel caso in cui desideri continuare a utilizzare Puppet Enterprise per le tue esigenze di gestione della configurazione al di fuori di. OpsWorks

Argomenti

- Passaggio 1: contatta Puppet per acquistare una licenza
- Passaggio 2: Ottieni dettagli sul tuo server OpsWorks for Puppet Enterprise
- Passaggio 3: Effettua un backup del tuo server OpsWorks for Puppet Enterprise
- Fase 4: Avviare una nuova istanza EC2
- Passaggio 5: installa Puppet Enterprise sulla nuova istanza EC2
- Passaggio 6: ripristinare il backup sulla nuova EC2 istanza
- Passaggio 7: configura la tua licenza Puppet
- Fase 8: Migrazione dei nodi
- Passaggio 9: Elimina il tuo server OpsWorks for Puppet Enterprise

Passaggio 1: contatta Puppet per acquistare una licenza

Quando esegui la migrazione dei server verso EC2, la nuova istanza non viene fornita con una licenza Puppet. Per acquistare una chiave di licenza, segui le istruzioni sul sito Web di Puppet.

Passaggio 2: Ottieni dettagli sul tuo server OpsWorks for Puppet Enterprise

Trova e salva i valori per il tuo server OpsWorks for Puppet Enterprise.

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

   Copia il nome del bucket Amazon S3 esistente per il tuo server for Puppet OpsWorks Enterprise. Il nome del bucket ha il formato: `aws-opsworks-cm-`*`server-name`*`-`*`random-string`*
2. Esegui il aws opsworks-cm describe-servers comando per ottenere la configurazione per il tuo server OpsWorks for Puppet Enterprise.

   ```
   aws opsworks-cm describe-servers \
      --server-name server-name \
      --region region
   ```

   Memorizza i valori per `InstanceTypeKeyPair`,`SubnetIds`,`SecurityGroupIds`,`InstanceProfileArn`, e `Endpoint` della risposta.
3. Usa SSH per connetterti al server OpsWorks for Puppet Enterprise esistente. Puoi usare Session Manager nella EC2 console anziché SSH.

   Esegui il comando seguente.

   ```
   rpm -qa | grep opsworks-cm-puppet-enterprise | cut -d '-' -f 5
   ```

   La risposta fornisce la versione di Puppet Enterprise (ad esempio, 2019.8.10). Memorizza questo valore.

Utilizzerai SSH o Session manager per il passaggio successivo.

Passaggio 3: Effettua un backup del tuo server OpsWorks for Puppet Enterprise

1. Esegui i seguenti comandi per effettuare un backup locale.

   ```
   mkdir /tmp/puppet-backup/
   sudo /opt/puppetlabs/bin/puppet-backup create --dir=/tmp/puppet-backup/
   ```
2. Esegui il comando seguente per memorizzare il nome per il backup.

   ```
   ls /tmp/puppet-backup/
   ```

```
PUPPET_BACKUP=$(ls /tmp/puppet-backup/)
```

3. Esegui il comando seguente per caricare il backup in un bucket S3. Sostituisci *S3-Bucket* con il valore del passaggio 1 in poi. Passaggio 2: Ottieni dettagli sul tuo server OpsWorks for Puppet Enterprise

   ```
   aws s3 cp /tmp/puppet-backup/PUPPET_BACKUP s3://S3_Bucket/tmp/puppet-backup/
   ```

   Memorizza S3_BUCKET i valori PUPPET_BACKUP and. Importerete questi valori nella nuova EC2 istanza.

È possibile uscire dalla sessione SSH o Session Manager.

Fase 4: Avviare una nuova istanza EC2

Avvia una nuova EC2 istanza dalla EC2 console https://console.aws.amazon.com/ec2/utilizzando la stessa configurazione del server OpsWorks for Puppet Enterprise.

Nome del parametro	Valore
SISTEMA OPERATIVO	Amazon Linux 2
Tipo di istanza	Il `InstanceType` valore del passaggio 2 di. Passaggio 2: Ottieni dettagli sul tuo server OpsWorks for Puppet Enterprise
Nome della coppia di chiavi	Il `KeyPair` valore del passaggio 2 diPassaggio 2: Ottieni dettagli sul tuo server OpsWorks for Puppet Enterprise.
VPC	Il VPC della fase 2 `SubnetIds` di. Passaggio 2: Ottieni dettagli sul tuo server OpsWorks for Puppet Enterprise
Sottorete	`SubnetIds` Dalla fase 2 di. Passaggio 2: Ottieni dettagli sul tuo server OpsWorks for Puppet Enterprise
Seleziona il gruppo di sicurezza esistente -> Gruppi di sicurezza comuni	`SecurityGroupIds` Dalla fase 2 diPassaggio 2: Ottieni dettagli sul tuo server OpsWorks for Puppet Enterprise.
Storage	Almeno 120 GB.

Nome del parametro	Valore
Profilo dell'istanza IAM	`InstanceProfileArn` Dalla fase 2 di[Passaggio 2: Ottieni dettagli sul tuo server OpsWorks for Puppet Enterprise]().

Se desideri creare e collegare un IP elastico alla nuova istanza, copia l'ID dell'istanza della nuova istanza e completa i passaggi indicati in[(Facoltativo) Fase 4.1: Creare e collegare un IP elastico]().

(Facoltativo) Fase 4.1: Creare e collegare un IP elastico

Mediante un indirizzo IP elastico, è possibile mascherare il guasto di un'istanza o di un software rimappando rapidamente l'indirizzo per un'altra istanza presente nell'account.

Per creare e associare un indirizzo IP elastico

1. Accedi a AWS Management Console e apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
2. Scegli Elastic IPs.
3. Scegli Alloca indirizzo IP elastico.
4. Dalla pagina Allocate Elastic IP address, scegli Allocate. Questo crea un indirizzo pubblico IPv4 .
5. Copia l' IPv4 indirizzo assegnato.
6. Da Azioni, scegli Associa indirizzo IP elastico.
7. Ad esempio, inserisci l'ID dell'istanza per la nuova istanza.
8. Selezionare Associate (Associa).

Passaggio 5: installa Puppet Enterprise sulla nuova istanza EC2

Usa SSH per connetterti alla nuova EC2 istanza. È possibile utilizzare Session Manager nella EC2 console anziché SSH.

```
# switch to sudo user
sudo -i

# Setup environment variables
PUPPET_ENTERPRISE_VERSION=Puppet Enterprise version from step 2.3
hostname Public IPv4 DNS or Custom Domain if available
```

```
# Install Puppet Enterprise
curl -JLO https://pm.puppetlabs.com/puppet-enterprise/$PUPPET_ENTERPRISE_VERSION/
puppet-enterprise-$PUPPET_ENTERPRISE_VERSION-el-7-x86_64.tar.gz
tar -xf puppet-enterprise-$PUPPET_ENTERPRISE_VERSION-el-7-x86_64.tar.gz

./puppet-enterprise-$PUPPET_ENTERPRISE_VERSION-el-7-x86_64/puppet-enterprise-installer
```

Puoi mantenere aperta la sessione SSH o Session Manager per il passaggio successivo.

Passaggio 6: ripristinare il backup sulla nuova EC2 istanza

```
# Setup environment variables
S3_BUCKET=S3 bucket name from step 2.1
PUPPET_BACKUP=Puppet backup file name from step 3.2

# download backup
aws s3 cp s3://$S3_BUCKET/tmp/puppet-backup/$PUPPET_BACKUP

# Prepare Puppet Enterprise backup to remove OpsWorks metadata
mkdir output
tar -xf $PUPPET_BACKUP -C output/
cd output/
rm -f opt/puppetlabs/facter/facts.d/opsworks.json
tar -cf ../$PUPPET_BACKUP *
cd ..
rm -rf output/

# Restore from backup
PATH=$PATH:/opt/puppetlabs/puppet/bin/
puppet-backup restore $PUPPET_BACKUP
puppet agent -t
```

Puoi accedere alla console Puppet per l' EC2 istanza ripristinata all'indirizzo https://*Public IPv4 of the instance*. Puoi trovare il IPv4 DNS pubblico nella pagina dei dettagli dell'istanza nella EC2 console. Le credenziali di accesso sono le stesse che usi per accedere al tuo server OpsWorks for Puppet Enterprise.

Puoi mantenere aperta la sessione SSH o Session Manager per il passaggio successivo.

Passaggio 7: configura la tua licenza Puppet

Segui i passaggi sul sito web di Puppet per configurare la tua licenza.

Puoi tenere aperta la sessione SSH o Session Manager per il passaggio successivo.

Fase 8: Migrazione dei nodi

Esistono due tipi di domini supportati dai quattro server Puppet OpsWorks Enterprise:

- BYODC (Bring Your Own Domain and Certificate)
- OpsWorks endpoint

Passaggio 8.1: Per BYODC (porta il tuo dominio e il tuo certificato)

Per questi nodi, tutto ciò che devi fare è indirizzare il dominio personalizzato del tuo provider DNS al DNS pubblico IPv4 o all' IPv4 indirizzo pubblico della nuova istanza. EC2

Fase 8.2: Per endpoint OpsWorks

Per un OpsWorks endpoint, la documentazione di Puppet consiglia di disinstallare l'agente Puppet sul nodo e quindi installare l'agente Puppet utilizzando il server Puppet Enterprise appena ripristinato.

> **Note**
>
> Sebbene Puppet non disponga di una procedura automatizzata per spostare un nodo agente, ci sono alcuni moduli che i membri della community di Puppet hanno pubblicato sul sito Web di Puppet Forge per eseguire la migrazione automatica dei nodi. Questi moduli includono il `pe_migrate`modulo e un secondo modulo di migrazione di un altro autore. I moduli sul sito web di Puppet Forge non sono supportati da Puppet o a OpsWorks meno che non sia esplicitamente indicato all'interno del modulo Forge. Consigliamo di usare cautela con questi moduli e di testarli prima di utilizzarli su larga scala.

Le seguenti sezioni forniscono i passaggi per disinstallare e reinstallare gli agenti Puppet sulle istanze Linux.

Argomenti

- Passaggio 8.2.1: Copia il programma di disinstallazione dal server Puppet
- Passaggio 8.2.2: scarica il programma di disinstallazione ed eseguilo su un nodo
- Passaggio 8.2.3: Reinstallare l'agente Puppet su un nodo

Passaggio 8.2.1: Copia il programma di disinstallazione dal server Puppet

Prima di disinstallare l'agente, assicurati che il profilo dell'istanza IAM del nodo fornisca le autorizzazioni S3. ReadOnly

Esegui il comando seguente per copiare il programma di disinstallazione dal server Puppet al bucket S3.

```
aws s3 cp \
   /opt/puppetlabs/bin/puppet-enterprise-uninstaller \
   s3://$S3_BUCKET/tmp/puppet-enterprise-uninstaller
```

Dopo aver eseguito il comando, puoi disconnetterti dalla sessione SSH o Session Manager del server Puppet.

Passaggio 8.2.2: scarica il programma di disinstallazione ed eseguilo su un nodo

Usa SSH per connetterti al nodo. È possibile utilizzare Session Manager nella EC2 console anziché SSH se il nodo è un' EC2 istanza.

```
sudo -i

S3_BUCKET=aws-opsworks-cm-abcdefg-uuhtyn6messn
aws s3 cp s3://$S3_BUCKET/tmp/puppet-enterprise-uninstaller /opt/puppetlabs/bin/
chmod 700 /opt/puppetlabs/bin/puppet-enterprise-uninstaller
/opt/puppetlabs/bin/puppet-enterprise-uninstaller
```

Puoi mantenere aperta la sessione SSH o Session Manager per il passaggio successivo.

Passaggio 8.2.3: Reinstallare l'agente Puppet su un nodo

Completa i seguenti passaggi per reinstallare l'agente Puppet su un nodo.

Argomenti

- Passaggio 8.2.3.1: Installa l'agente Puppet con la configurazione corretta
- Passaggio 8.2.3.2: Accetta il certificato nella console Puppet
- Passaggio 8.2.3.3: Controlla il nodo nel server Puppet Enterprise

Passaggio 8.2.3.1: Installa l'agente Puppet con la configurazione corretta

Esegui il comando seguente per installare l'agente Puppet.

```
curl -k https://Public_IPv4_DNS:8140/packages/current/install.bash | bash
```

Puoi mantenere aperta la sessione SSH o Session Manager per il passaggio 8.2.2.3.

Passaggio 8.2.3.2: Accetta il certificato nella console Puppet

1. Vai alla console del server Puppet all'indirizzo. `https://Public_IPv4_DNS`
2. Scegli Certificati, quindi Certificati non firmati.
3. Scegli Accetta per firmare il certificato dell'agente Puppet.

Passaggio 8.2.3.3: Controlla il nodo nel server Puppet Enterprise

Esegui il seguente comando sul nodo per archiviarlo nel server.

```
puppet agent -t
```

Il nodo dovrebbe ora essere visibile nella console del server Puppet.

Passaggio 9: Elimina il tuo server OpsWorks for Puppet Enterprise

Puoi utilizzare la OpsWorks console o AWS CLI eliminare il tuo server OpsWorks for Puppet Enterprise.

Per eliminare il server utilizzando la console OpsWorks

1. Accedi a AWS Management Console e apri la AWS OpsWorks console all'indirizzo https://console.aws.amazon.com/opsworks/.
2. Scegli i server Puppet Enterprise dal pannello di navigazione.
3. Nella pagina dei server Puppet Enterprise, scegli il server che desideri eliminare.
4. Da Azioni, scegli Elimina server Puppet Enterprise.

Per eliminare il server utilizzando il AWS CLI

Esegui il comando seguente.

```
aws opsworks-cm delete-server \
   --server-name server-name \
   --region region
```

Registrazione OpsWorks per le chiamate API Puppet Enterprise con AWS CloudTrail

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

OpsWorks for Puppet Enterprise è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in OpsWorks for Puppet Enterprise. CloudTrail acquisisce tutte le chiamate API OpsWorks per Puppet Enterprise come eventi, incluse le chiamate dalla console OpsWorks for Puppet Enterprise e le chiamate di codice a for Puppet Enterprise. OpsWorks APIs Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi OpsWorks per Puppet Enterprise. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta a OpsWorks cui è stata inviata Puppet Enterprise, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida per l'AWS CloudTrail utente.

OpsWorks per informazioni su Puppet Enterprise in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in OpsWorks for Puppet Enterprise, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi OpsWorks per Puppet Enterprise, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un trail nella console, il trail sarà valido in tutte le regioni. Il trail registra gli eventi da tutte le regioni della AWS partizione e consegna i file di log al

bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consultare:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte OpsWorks le azioni di Puppet Enterprise vengono registrate CloudTrail e documentate nel riferimento all'API OpsWorks for Puppet Enterprise. Ad esempio, le chiamate alle DescribeServersazioni CreateServerCreateBackup, e generano voci nei file di registro. CloudTrail

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali dell'utente IAM o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Comprensione OpsWorks delle voci dei file di registro di Puppet Enterprise

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro per l'azione OpsWorks for Puppet Enterprise`CreateServer`.

```
{"eventVersion":"1.05",
"userIdentity":{
    "type":"AssumedRole",
```

```
    "principalId":"ID number:OpsWorksCMUser",
    "arn":"arn:aws:sts::831000000000:assumed-role/Admin/OpsWorksCMUser",
    "accountId":"831000000000","accessKeyId":"ID number",
    "sessionContext":{
        "attributes":{
            "mfaAuthenticated":"false",
            "creationDate":"2017-01-05T22:03:47Z"
            },
        "sessionIssuer":{
            "type":"Role",
            "principalId":"ID number",
            "arn":"arn:aws:iam::831000000000:role/Admin",
            "accountId":"831000000000",
            "userName":"Admin"
            }
        }
    },
"eventTime":"2017-01-05T22:18:23Z",
"eventSource":"opsworks-cm.amazonaws.com",
"eventName":"CreateServer",
"awsRegion":"us-west-2",
"sourceIPAddress":"101.25.190.51",
"userAgent":"console.amazonaws.com",
"requestParameters":{
    "serverName":"test-puppet-server",
    "engineModel":"Single",
    "engine":"Puppet",
    "instanceProfileArn":"arn:aws:iam::831000000000:instance-profile/aws-opsworks-cm-
ec2-role",
    "backupRetentionCount":3,"serviceRoleArn":"arn:aws:iam::831000000000:role/service-
role/aws-opsworks-cm-service-role",
    "engineVersion":"12",
    "preferredMaintenanceWindow":"Fri:21:00",
    "instanceType":"t2.medium",
    "subnetIds":["subnet-1e111f11"],
    "preferredBackupWindow":"Wed:08:00"
    },
"responseElements":{
    "server":{
        "endpoint":"test-puppet-server-xxxx8u4390xo6pd9.us-west-2.opsworks-cm.io",
        "createdAt":"Jan 5, 2017 10:18:22 PM",
        "serviceRoleArn":"arn:aws:iam::831000000000:role/service-role/aws-opsworks-cm-
service-role",
        "preferredBackupWindow":"Wed:08:00",
```

```
        "status":"CREATING",
        "subnetIds":["subnet-1e111f11"],
        "engine":"Puppet",
        "instanceType":"t2.medium",
        "serverName":"test-puppet-server",
        "serverArn":"arn:aws:opsworks-cm:us-west-2:831000000000:server/test-puppet-
server/8ezz7f6z-e91f-4z10-89z5-8c6219zzz09f",
        "engineModel":"Single",
        "backupRetentionCount":3,
        "engineAttributes":[
            {"name":"PUPPET_ADMIN_PASSWORD","value":"*** Redacted ***"},
            {"name":"PUPPET_API_CA_CERT","value":"*** Redacted ***"},
            ],
        "engineVersion":"12.11.1",
        "instanceProfileArn":"arn:aws:iam::831000000000:instance-profile/aws-opsworks-
cm-ec2-role",
        "preferredMaintenanceWindow":"Fri:21:00"
        }
    },
"requestID":"de7z64z9-d394-12ug-8081-7zz0386fbcb6",
"eventID":"8z7z18dz-6z90-47bz-87cf-e8346428zzz3",
"eventType":"AwsApiCall",
"recipientAccountId":"831000000000"
}
```


Risoluzione dei problemi OpsWorks per Puppet Enterprise

> **Important**
>
> Il AWS OpsWorks for Puppet Enterprise servizio ha raggiunto la fine del ciclo di vita il 31 marzo 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo argomento contiene alcuni problemi comuni relativi OpsWorks a Puppet Enterprise e le soluzioni suggerite per tali problemi.

Argomenti

- Suggerimenti generali per la risoluzione dei problemi
- Risoluzione di errori specifici
- Supporto e informazioni della guida aggiuntivi

Suggerimenti generali per la risoluzione dei problemi

Se non sei in grado di creare o utilizzare un master Puppet, puoi visualizzare i messaggi di errore o i log per risolvere il problema. Le attività seguenti descrivono da dove iniziare per affrontare la risoluzione dei problemi relativi a un master Puppet. Per informazioni su errori e soluzioni specifici, consulta la sezione Risoluzione di errori specifici di questo argomento.

- Usa la console OpsWorks for Puppet Enterprise per visualizzare i messaggi di errore se un Puppet master non si avvia. Nella pagina delle proprietà del master Puppet, i messaggi di errore relativi all'avvio e all'esecuzione del server vengono visualizzati nella parte superiore della pagina. Gli errori possono derivare dai OpsWorks servizi Puppet Enterprise o Amazon EC2 utilizzati per creare un Puppet master. AWS CloudFormation Nella pagina delle proprietà puoi anche visualizzare gli eventi che si verificano su un server in esecuzione e che possono contenere messaggi relativi agli eventi di errore.
- Per risolvere i EC2 problemi, connettiti all'istanza del server tramite SSH e visualizza i log. EC2 i log delle istanze vengono archiviati nella directory. `/var/log/aws/opsworks-cm` Questi registri acquisiscono gli output dei comandi mentre OpsWorks per Puppet Enterprise avvia un Puppet master.

Risoluzione di errori specifici

Argomenti

- Il server si trova in uno stato di interruzione della connessione
- Creazione del server non riuscita e visualizzazione messaggio "requested configuration is currently not supported" (configurazione richiesta attualmente non supportata)
- Impossibile creare l' EC2 istanza Amazon del server
- Un errore del ruolo del servizio impedisce la creazione del server
- Superamento del limite di indirizzi IP elastici
- Associazione automatica dei nodi non riuscita
- La manutenzione del sistema fallisce

Il server si trova in uno stato di interruzione della connessione

Problema: lo stato di un server viene visualizzato come Connessione interrotta.

Causa: ciò si verifica più comunemente quando un'entità esterna AWS OpsWorks apporta modifiche a un server OpsWorks for Puppet Enterprise o alle relative risorse di supporto. AWS OpsWorks non può connettersi ai server Puppet Enterprise nello stato Connection lost per gestire attività di manutenzione come la creazione di backup, l'applicazione di patch al sistema operativo o l'aggiornamento di Puppet. Di conseguenza, sul server potrebbero mancare aggiornamenti importanti, presentare problemi di sicurezza o comunque non funzionare come previsto.

Soluzione: prova i seguenti passaggi per ripristinare la connessione del server.

1. Assicurati che il tuo ruolo di servizio disponga di tutte le autorizzazioni necessarie.
    a. Nella pagina Impostazioni del server, in Rete e sicurezza, scegli il link per il ruolo di servizio utilizzato dal server. Questo apre il ruolo di servizio per la visualizzazione nella console IAM.
    b. Nella scheda Autorizzazioni, verifica che `AWSOpsWorksCMServiceRole` sia presente nell'elenco delle politiche di autorizzazione. Se non è elencata, aggiungi manualmente la politica `AWSOpsWorksCMServiceRole` gestita al ruolo.
    c. Nella scheda Relazioni di fiducia, verifica che il ruolo di servizio disponga di una politica di fiducia che preveda che il `opsworks-cm.amazonaws.com` servizio assuma ruoli per tuo conto. Per ulteriori informazioni su come utilizzare le policy di fiducia con i ruoli, consulta Modifying a role (console) o il post del AWS Security Blog, How to use trust policies with IAM roles.
2. Assicurati che il profilo dell'istanza disponga di tutte le autorizzazioni necessarie.
    a. Nella pagina Impostazioni del server, in Rete e sicurezza, scegli il link per il profilo di istanza utilizzato dal server. Questo apre il profilo dell'istanza per la visualizzazione nella console IAM.
    b. Nella scheda Autorizzazioni, verifica che `AmazonEC2RoleforSSM` `AWSOpsWorksCMInstanceProfileRole` siano entrambe incluse nell'elenco delle politiche di autorizzazione. Se una o entrambe non sono elencate, aggiungi queste politiche gestite manualmente al ruolo.
    c. Nella scheda Relazioni di fiducia, verifica che il ruolo di servizio disponga di una politica di fiducia che preveda che il `ec2.amazonaws.com` servizio assuma ruoli per tuo conto. Per ulteriori informazioni su come utilizzare le policy di fiducia con i ruoli, consulta Modifying a role (console) o il post del AWS Security Blog, How to use trust policies with IAM roles.

3. Nella EC2 console Amazon, assicurati di trovarti nella stessa regione del server OpsWorks for Puppet Enterprise, quindi riavvia l' EC2 istanza utilizzata dal server.

   a. Scegli l' EC2 istanza `aws-opsworks-cm-instance-` *`server-name`* denominata.

   b. Nel menu Stato dell'istanza, scegli Riavvia istanza.

   c. Attendi fino a 15 minuti prima che il server si riavvii e sia completamente online.

4. Nella console OpsWorks for Puppet Enterprise, nella pagina dei dettagli del server, verifica che lo stato del server sia ora integro.

Se lo stato del server è ancora Connessione interrotta dopo aver eseguito i passaggi precedenti, prova una delle seguenti soluzioni.

- Sostituisci il server creandone uno nuovo ed eliminando l'originale. Se i dati sul server corrente sono importanti per te, ripristina il server da un backup recente e verifica che i dati siano aggiornati prima di eliminare il server originale che non risponde.
- Contatta l'assistenza. AWS

Creazione del server non riuscita e visualizzazione messaggio "requested configuration is currently not supported" (configurazione richiesta attualmente non supportata)

Problema: stai cercando di creare un server Puppet Enterprise, ma la creazione del server ha esito negativo e viene visualizzato un messaggio indicante che la configurazione richiesta attualmente non è supportata. Viene inoltre richiesto di controllare le configurazioni supportate nella documentazione.

Causa: è possibile che sia stato specificato un tipo di istanza non supportato per il master Puppet. Se scegli di creare il server Puppet in un VPC con una tenancy non predefinita, ad esempio una per le istanze dedicate, anche tutte le istanze all'interno del VPC specificato devono avere un'istanza dedicata a tenant o host singolo. Poiché alcuni tipi di istanza, ad esempio t2, sono disponibili solo con la tenancy predefinita, il tipo di istanza del master Puppet potrebbe non essere supportato dal VPC specificato e pertanto la creazione del server ha esito negativo.

Soluzione: se scegli un VPC con una tenancy non predefinita, utilizza un tipo di istanza m4, che può supportare l'istanza dedicata a tenant singolo.

Impossibile creare l' EC2 istanza Amazon del server

Problema: la creazione del server non è riuscita con un messaggio di errore simile al seguente: «Impossibile creare le seguenti risorse: [EC2Istanza]. Failed to receive 1 resource signal(s) within the specified duration." (Impossibile creare la seguente o le seguenti risorse: [EC2Instance]. Impossibile ricevere un segnale risorsa nell'intervallo specificato.)

Causa: Ciò è probabilmente dovuto al fatto che l' EC2istanza non ha accesso alla rete.

Soluzione: assicurati che l'istanza disponga dell'accesso a Internet in uscita e che l'agente AWS di servizio sia in grado di emettere comandi. Verifica che per il VPC (un VPC con una singola sottorete pubblica) sia stata abilitata la funzionalità DNS resolution (Risoluzione DNS) e che per la sottorete sia stata abilitata l'impostazione Auto-assign Public IP (Assegna automaticamente IP pubblico).

Un errore del ruolo del servizio impedisce la creazione del server

Problema: la creazione del server non riesce e viene visualizzato un messaggio di errore che indica «Non autorizzato a eseguire sts:AssumeRole».

Causa: questa situazione si può verificare quando il ruolo del servizio in uso non dispone delle autorizzazioni adeguate per creare un nuovo server.

Soluzione: apri la console OpsWorks for Puppet Enterprise; usa la console per generare un nuovo ruolo di servizio e un ruolo del profilo di istanza. Se preferisci utilizzare il tuo ruolo di servizio, allega la policy AWSOpsWorks CMService Role al ruolo. Verifica che opsworks-cm.amazonaws.com sia elencato tra i servizi nelle relazioni di fiducia del ruolo. Verifica che al ruolo di servizio associato al Puppet master sia allegata la politica gestita da Works Role. AWSOps CMService

Superamento del limite di indirizzi IP elastici

Problema: la creazione del server non riesce e viene visualizzato un messaggio di errore che indica: «Impossibile creare le seguenti risorse: [EIP, EC2 Istanza]. Resource creation cancelled, the maximum number of addresses has been reached." (Impossibile creare la seguente o le seguenti risorse: [EIP, EC2Instance]. Creazione della risorsa annullata. È stato raggiunto il numero massimo di indirizzi)

Causa: questo potrebbe verificarsi quando l'account ha utilizzato il numero massimo di indirizzi IP elastici (EIP). Il limite predefinito di indirizzi EIP è cinque.

Soluzione: puoi rilasciare gli indirizzi EIP esistenti o eliminare quelli che il tuo account non utilizza attivamente oppure puoi contattare l'Assistenza AWS clienti per aumentare il limite di indirizzi EIP associato al tuo account.

Associazione automatica dei nodi non riuscita

Problema: l'associazione automatica o incustodita di nuovi EC2 nodi Amazon non funziona. I nodi che dovrebbero venire aggiunti al master Puppet non vengono visualizzati nel pannello di controllo di Puppet Enterprise.

Causa: ciò può verificarsi quando non è impostato un ruolo IAM come profilo di istanza che consente alle chiamate `opsworks-cm` API di comunicare con nuove istanze. EC2

Soluzione: allega una policy al profilo dell' EC2 istanza che consenta di `DescribeNodeAssociationStatus` utilizzare `AssociateNode` le chiamate API EC2, come descritto in. Aggiungere nodi automaticamente in OpsWorks Puppet Enterprise

La manutenzione del sistema fallisce

AWS OpsWorks CM esegue la manutenzione settimanale del sistema per garantire che le ultime versioni AWS testate di Puppet Server, inclusi gli aggiornamenti di sicurezza, siano sempre in esecuzione su un server OpsWorks for Puppet Enterprise. Se, per qualsiasi motivo, la manutenzione del sistema fallisce, ti AWS OpsWorks CM avvisa dell'errore. Per ulteriori informazioni sulla manutenzione del sistema, vedereManutenzione del sistema OpsWorks per Puppet Enterprise.

Questa sezione descrive i possibili motivi di errore e suggerisce soluzioni.

Argomenti

- Un errore nel profilo del ruolo o dell'istanza del servizio impedisce la manutenzione del sistema

Un errore nel profilo del ruolo o dell'istanza del servizio impedisce la manutenzione del sistema

Problema: la manutenzione del sistema non riesce con un messaggio di errore che indica «Non autorizzato a eseguire sts: AssumeRole «o un messaggio di errore simile relativo alle autorizzazioni.

Causa: ciò può verificarsi quando il ruolo di servizio o il profilo di istanza utilizzato non dispongono delle autorizzazioni adeguate per eseguire la manutenzione del sistema sul server.

Soluzione: assicurati che il ruolo di servizio e il profilo di istanza dispongano di tutte le autorizzazioni necessarie.

1. Assicurati che il tuo ruolo di servizio disponga di tutte le autorizzazioni necessarie.

   a. Nella pagina Impostazioni del server, in Rete e sicurezza, scegli il link per il ruolo di servizio utilizzato dal server. Questo apre il ruolo di servizio per la visualizzazione nella console IAM.

   b. Nella scheda Autorizzazioni, verifica che `AWSOpsWorksCMServiceRole` sia associato al ruolo di servizio. Se non `AWSOpsWorksCMServiceRole` è elencato, aggiungi questo criterio al ruolo.

   c. Verifica che opsworks-cm.amazonaws.com sia elencato tra i servizi nelle relazioni di fiducia del ruolo. Per ulteriori informazioni su come utilizzare le politiche di fiducia con i ruoli, consulta Modifying a role (console) o il post del AWS Security Blog, How to use trust policies with IAM roles.

2. Assicurati che il profilo dell'istanza disponga di tutte le autorizzazioni necessarie.

   a. Nella pagina Impostazioni del server, in Rete e sicurezza, scegli il link per il profilo di istanza utilizzato dal server. Questo apre il profilo dell'istanza per la visualizzazione nella console IAM.

   b. Nella scheda Autorizzazioni, verifica che `AmazonEC2RoleforSSM` `AWSOpsWorksCMInstanceProfileRole` siano entrambe incluse nell'elenco delle politiche di autorizzazione. Se una o entrambe non sono elencate, aggiungi queste politiche gestite manualmente al ruolo.

   c. Nella scheda Relazioni di fiducia, verifica che il ruolo di servizio disponga di una politica di fiducia che preveda che il `ec2.amazonaws.com` servizio assuma ruoli per tuo conto. Per ulteriori informazioni su come utilizzare le policy di fiducia con i ruoli, consulta Modifying a role (console) o il post del AWS Security Blog, How to use trust policies with IAM roles.

Supporto e informazioni della guida aggiuntivi

Se la risoluzione di un problema specifico non è descritta in questo argomento oppure se hai provato i suggerimenti riportati in questo argomento ma il problema persiste, visita i forum di AWS OpsWorks.

Puoi anche visitare il Centro AWS Support. Il AWS Support Center è l'hub per la creazione e la gestione dei casi di AWS assistenza. Il AWS Support Center include anche collegamenti ad altre risorse utili, come forum, informazioni tecniche FAQs, stato di salute del servizio e AWS Trusted Advisor.

AWS OpsWorks per Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

AWS OpsWorks for Chef Automate consente di eseguire un server Chef Automate in AWS. È possibile effettuare il provisioning di un server Chef in pochi minuti e lasciare che se ne AWS OpsWorks for Chef Automate occupi le operazioni, i backup, i ripristini e gli aggiornamenti software. AWS OpsWorks for Chef Automate ti consente di concentrarti sulle attività principali di gestione della configurazione, invece di gestire un server Chef.

Un server Chef Automate gestisce la configurazione dei nodi nell'ambiente indicando `chef-client`quali ricette Chef eseguire sui nodi, archivia le informazioni sui nodi e funge da archivio centrale per i libri di cucina Chef. AWS OpsWorks for Chef Automate fornisce server Chef che includono funzionalità premium di Chef Automate: Chef Infra e Chef. InSpec

Un AWS OpsWorks for Chef Automate server viene eseguito su un'istanza Amazon Elastic Compute Cloud. AWS OpsWorks for Chef Automate i server sono configurati per eseguire la versione più recente di Amazon Linux (Amazon Linux 2). Per informazioni sulle modifiche apportate a questa versione di Chef Automate, consulta le Note di rilascio di Chef Automate. La tabella seguente descrive i componenti Chef installati su un AWS OpsWorks for Chef Automate server.

Nome componente	Descrizione	Versione installata sul AWS OpsWorks for Chef Automate server
Chef Automate	Chef Automate è un pacchetto software server di livello enterprise che fornisce un flusso di lavoro automatizzato per la distribuzione continua e approfond	2.0

Nome componente	Descrizione	Versione installata sul AWS OpsWorks for Chef Automate server
	imenti sui nodi gestiti in una console di gestione basata sul Web. Chef Automate offre l'automazione dell'infrastruttura includendo Chef Infra, le informazioni e l'applicazione della sicurezza e della conformità includendo Chef InSpec e l'implementazione automatizzata includendo Chef Habitat. Per ulteriori informazioni su Chef Automate, consulta Chef Automate nel sito Web di Chef.	
Chef Infra	Precedentemente chiamato Chef Server, Chef Infra Server usa l'agente Chef Infra Client (`chef-client`) per applicare in modo continuo le configurazioni ai nodi gestiti per mantenere uno stato desiderato. Per ulteriori informazioni su Infra, consulta Chef Infra nel sito Web di Chef.	12.x

Nome componente	Descrizione	Versione installata sul AWS OpsWorks for Chef Automate server
Chef InSpec	Chef InSpec descrive le regole di sicurezza e conformità che possono essere condivise tra ingegneri del software, tecnici operativi e tecnici della sicurezza. Conformità, sicurezza e altri requisiti delle policy costituis cono il framework per i test automatizzati che l'agente `chef-client` può eseguire sui nodi gestiti, garantend o un'applicazione coerente degli standard. Per ulteriori informazioni in merito InSpec, consulta Chef InSpec sul sito Web di Chef.	3.9.0

La versione minima supportata di `chef-client` sui nodi associati a un server AWS OpsWorks for Chef Automate è la versione 13.x. Ti consigliamo di eseguire almeno la versione 14.10.9 o la versione stabile `chef-client` più recente.

Quando le versioni secondarie del software Chef diventano disponibili, la manutenzione del sistema viene progettata per aggiornare automaticamente la versione secondaria di Chef Automate e Chef Server sul server, non appena supera il test AWS. AWS esegue test approfonditi per verificare che gli upgrade di Chef siano pronti per la produzione e non interferiscano con gli ambienti dei clienti esistenti, quindi possono verificarsi ritardi tra le versioni del software Chef e la loro disponibilità per l'applicazione sui server Chef Automate esistenti OpsWorks. La manutenzione del sistema aggiorna inoltre il server in base alla versione più recente di Amazon Linux.

Puoi connettere qualsiasi computer o EC2 istanza locale che esegua un sistema operativo supportato e abbia accesso di rete a un server. AWS OpsWorks for Chef Automate Per un elenco dei sistemi

operativi supportati per i nodi che si desidera gestire, consulta il sito Web di Chef. Il software dell'agente `chef-client` viene installato sui nodi che si desidera gestire con un server Chef.

Argomenti

- Region Support per AWS OpsWorks for Chef Automate
- AWS OpsWorks per la fine del ciclo di vita di Chef Automate FAQs
- Aggiorna un AWS OpsWorks for Chef Automate server a Chef Automate 2
- Guida introduttiva con AWS OpsWorks for Chef Automate
- Creare un AWS OpsWorks for Chef Automate server utilizzando AWS CloudFormation
- Aggiornare un AWS OpsWorks for Chef Automate server per utilizzare un dominio personalizzato
- Rigenera lo starter kit per un server AWS OpsWorks for Chef Automate
- Lavorare con i tag sulle AWS OpsWorks for Chef Automate risorse
- Backup e ripristino di un AWS OpsWorks for Chef Automate server
- Manutenzione del sistema in AWS OpsWorks for Chef Automate
- Scansioni di conformità in AWS OpsWorks for Chef Automate
- Dissociare un nodo da un server AWS OpsWorks for Chef Automate
- Eliminare un AWS OpsWorks for Chef Automate server
- Reimpostazione delle credenziali del pannello di controllo di Chef Automate
- Registrazione delle chiamate AWS OpsWorks for Chef Automate API con AWS CloudTrail
- Risoluzione dei problemi AWS OpsWorks for Chef Automate

Region Support per AWS OpsWorks for Chef Automate

I seguenti endpoint regionali supportano i AWS OpsWorks for Chef Automate server. AWS OpsWorks for Chef Automate crea risorse associate ai server Chef, come profili di istanza, utenti e ruoli di servizio, nello stesso endpoint regionale del server Chef. Il server Chef deve essere in un VPC. È possibile utilizzare un VPC che si crea o di cui si dispone già, oppure utilizzare il VPC predefinito.

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Regione Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)

- Regione Asia Pacifico (Tokyo)
- Regione Asia Pacifico (Singapore)
- Regione Asia Pacifico (Sydney)
- Regione Europa (Francoforte)
- Regione Europa (Irlanda)

AWS OpsWorks per la fine del ciclo di vita di Chef Automate FAQs

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa.

Argomenti

- In che modo gli utenti esistenti saranno influenzati da questa fine del ciclo di vita?
- Cosa succede ai miei server se non eseguo alcuna azione?
- A quali alternative posso passare?
- Il servizio accetta ancora nuovi clienti?
- La fine del ciclo di vita influirà Regioni AWS su tutti allo stesso tempo?
- Che livello di supporto tecnico è disponibile?
- Sono un cliente attuale di OpsWorks for Chef Automate e ho bisogno di avviare un server in un account che in precedenza non utilizzava il servizio. Sono in grado di farlo?
- Ci saranno importanti novità nel corso del prossimo anno?

In che modo gli utenti esistenti saranno influenzati da questa fine del ciclo di vita?

I clienti esistenti non saranno interessati fino al 5 maggio 2024, data di fine del ciclo di vita di Chef OpsWorks Automate. Dopo la data di fine del ciclo di vita, i clienti non saranno più in grado di gestire i propri server utilizzando la OpsWorks console o l'API.

Cosa succede ai miei server se non eseguo alcuna azione?

A partire dal 5 maggio 2024, non sarai più in grado di gestire i tuoi server utilizzando la OpsWorks console o l'API. A quel punto, smetteremo di eseguire qualsiasi funzione di gestione continua per i tuoi server, come i backup o la manutenzione. Per limitare l'impatto sui clienti, lasceremo in funzione tutte EC2 le istanze che eseguono il backup dei server Chef Automate, ma le relative licenze non saranno più valide in quanto l'utilizzo non è più coperto (o fatturato) dal contratto di servizio for Chef Automate con Chef. OpsWorks Dovrai contattare Chef per ottenere una nuova licenza. Quando contatti Chef, assicurati di dire loro che sei un cliente esistente di Chef Automate e che stai OpsWorks per passare da. OpsWorks

A quali alternative posso passare?

AWS e Progress Chef ti consigliano di migrare alla loro nuova offerta Chef SaaS in modo da poter continuare a beneficiare di un servizio Chef Automate completamente gestito. Per iniziare con Chef SaaS, puoi contattare Chef per ottenere la documentazione su come configurare un account Chef SaaS e trasferire dati e nodi.

Se Chef SaaS non soddisfa le tue esigenze perché preferisci eseguire Chef Automate su EC2 istanze AWS degli account che controlli, Chef offre diverse opzioni tra cui un modello Marketplace AWS Bring Your Own License (BYOL) e l'hosting autonomo. EC2 Puoi contattare Progress Chef per ulteriori informazioni su come eseguire tale transizione.

Il servizio accetta ancora nuovi clienti?

No. AWS OpsWorks for Chef Automate non accetta più nuovi clienti.

La fine del ciclo di vita influirà Regioni AWS su tutti allo stesso tempo?

Sì. L'API e la console raggiungeranno la fine del loro ciclo di vita e saranno completamente inutilizzabili a partire dal 5 maggio 2024. Regioni AWS Per informazioni su Regioni AWS dove AWS OpsWorks è disponibile Chef Automate, consulta l'Elenco dei servizi AWS regionali.

Che livello di supporto tecnico è disponibile?

AWS continuerà a fornire a Chef Automate lo stesso livello OpsWorks di supporto che i clienti hanno oggi fino alla data di fine del ciclo di vita. Se hai domande o dubbi, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support. Per il supporto alla transizione, consigliamo ai clienti di contattare Progress Chef.

Sono un cliente attuale di OpsWorks for Chef Automate e ho bisogno di avviare un server in un account che in precedenza non utilizzava il servizio. Sono in grado di farlo?

Generalmente no, a meno che non vi siano circostanze eccezionali per farlo. Se hai una situazione particolare, contatta il Supporto AWS Team su AWS Re:post o tramite Premium AWS Support con i dettagli e la giustificazione e esamineremo la tua richiesta.

Ci saranno importanti novità nel corso del prossimo anno?

No. Poiché il servizio sta raggiungendo la fine del ciclo di vita, non rilasceremo alcuna nuova funzionalità. Tuttavia, continueremo a migliorare la sicurezza e a gestire i server come previsto fino alla data di fine del ciclo di vita.

Aggiorna un AWS OpsWorks for Chef Automate server a Chef Automate 2

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Prerequisiti per l'aggiornamento a Chef Automate 2

Prima di iniziare, assicurati di aver compreso le nuove funzionalità aggiunte in Chef Automate 2 e le funzionalità che Chef Automate 2 non supporta. Per informazioni sulle funzionalità nuove e su quelle non supportate di Chef Automate 2, consulta la documentazione di Chef Automate 2 sul sito Web di Chef.

Un server che esegue Chef Automate 1 deve aver completato almeno una manutenzione dopo il 1° novembre 2019 per essere idoneo all'aggiornamento.

Come per qualsiasi operazione di manutenzione sul AWS OpsWorks for Chef Automate server, il server è offline durante l'aggiornamento. Ti consigliamo di pianificare fino a tre ore di inattività per il processo di aggiornamento.

Sono necessarie le credenziali di accesso per questo server per il sito Web del pannello di controllo di Chef Automate. Al termine dell'aggiornamento, devi accedere al pannello di controllo di Chef Automate e verificare che i nodi e le informazioni di configurazione non siano cambiati.

> **Important**
>
> Quando sei pronto per aggiornare il tuo AWS OpsWorks for Chef Automate server a Chef Automate 2, usa solo le istruzioni qui per l'aggiornamento. Poiché AWS OpsWorks for Chef Automate automatizza molti dei processi di aggiornamento, come la creazione di backup, non seguite le istruzioni di aggiornamento sul sito Web di Chef.

Informazioni sul processo di aggiornamento

Durante il processo di aggiornamento, il server viene sottoposto al backup prima di iniziare l'aggiornamento e dopo aver terminato l'aggiornamento. Vengono creati i seguenti backup:

- Un backup del server quando è ancora in esecuzione Chef Automate 1 (versione 12.17.33).
- Un backup del server al termine dell'aggiornamento che esegue Chef Automate 2 (versione 2019-08).

Il processo di aggiornamento termina l' EC2 istanza Amazon utilizzata dal server quando eseguiva Chef Automate 1. Viene creata una nuova istanza per eseguire il server Chef Automate 2.

Aggiornamento a Chef Automate 2 (console)

1. Accedi a AWS Management Console e apri la AWS OpsWorks console all'indirizzo. https://console.aws.amazon.com/opsworks/
2. Nel riquadro di navigazione a sinistra, scegliere AWS OpsWorks for Chef Automate.
3. Scegliere un server per visualizzarne la pagina delle proprietà. Un banner blu nella parte superiore della pagina indica se il server è idoneo per l'aggiornamento a Chef Automate 2.

> **Note**
>
> Un server che esegue Chef Automate 1 deve aver completato almeno una manutenzione dopo il 1° novembre 2019 per essere idoneo all'aggiornamento.

4. Se il server è idoneo per l'aggiornamento, scegliere Start upgrade (Avvia aggiornamento).
5. Consentire fino a tre ore per l'aggiornamento. Durante il processo di aggiornamento, nella pagina delle proprietà è visualizzato lo stato del server come Under maintenance (In manutenzione).
6. Al termine dell'aggiornamento, nella pagina delle proprietà vengono visualizzati i due messaggi seguenti: Successfully upgraded to Automate 2 (Aggiornamento ad Automate 2 completato e Maintenance completed successfully (Manutenzione completata). Lo stato del server deve essere HEALTHY (INTEGRO).
7. Accedere al pannello di controllo Chef Automate con le credenziali esistenti e verificare che i nodi stiano funzionando correttamente.

Aggiornamento a Chef Automate 2 (CLI)

1. (Facoltativo) Se non sei sicuro di quali dei tuoi AWS OpsWorks for Chef Automate server siano idonei all'aggiornamento, esegui il comando seguente. Assicurati di aggiungere il `--region` parametro se desideri elencare AWS OpsWorks for Chef Automate i server in una regione AWS diversa dalla tua regione AWS predefinita.

   ```
   aws opsworks-cm describe-servers
   ```

   Nei risultati, cercare il valore `true` per l'attributo `CHEF_MAJOR_UPGRADE_AVAILABLE`. Ciò indica che il server è idoneo per l'aggiornamento a Chef Automate 2. Prendi nota dei nomi dei AWS OpsWorks for Chef Automate server idonei per l'aggiornamento.
2. Esegui il comando seguente, sostituendolo *`server_name`* con il nome di un AWS OpsWorks for Chef Automate server. Per eseguire l'aggiornamento a Chef Automate 2 invece di eseguire la manutenzione ordinaria del sistema, aggiungere l'attributo del motore `CHEF_MAJOR_UPGRADE`, come mostrato nel comando. Aggiungere il parametro `--region` se il server di destinazione non è nella regione AWS predefinita. È possibile aggiornare un solo server per comando.

   ```
   aws opsworks-cm start-maintenance --server-name server_name --engine-attributes
    Name=CHEF_MAJOR_UPGRADE,Value=true --region region
   ```

Se AWS OpsWorks for Chef Automate non è possibile aggiornare il server per qualsiasi motivo, questo comando genera un'eccezione di convalida.

3. Consentire fino a tre ore per l'aggiornamento. È possibile controllare periodicamente lo stato dell'aggiornamento eseguendo il comando seguente.

```
aws opsworks-cm describe-servers --server-name server_name
```

Nei risultati, cercare il valore `Status`. Lo `Status` `UNDER_MAINTENANCE` indica che l'aggiornamento è ancora in corso. Un aggiornamento completato restituisce messaggi simili ai seguenti.

```
2019/10/24 00:27:56 UTC          Successfully upgraded to Automate 2.
2019/10/23 23:50:38 UTC          Upgrading Chef server from Automate 1 to Automate
 2
```

Se l'aggiornamento non ha avuto successo, ripristina AWS OpsWorks for Chef Automate automaticamente il server a Chef Automate 1.

Se l'aggiornamento ha avuto esito positivo ma il server non funziona come prima dell'aggiornamento (ad esempio, se i nodi gestiti non funzionano), è possibile ripristinare manualmente il server. Per informazioni sul rollback manuale, vedere Ripristina un AWS OpsWorks for Chef Automate server su Chef Automate 1 (CLI).

Ripristina un AWS OpsWorks for Chef Automate server su Chef Automate 1 (CLI)

Se il processo di aggiornamento fallisce, ripristina AWS OpsWorks for Chef Automate automaticamente il server a Chef Automate 1. Se l'aggiornamento è andato a buon fine ma il server non funziona come prima dell'aggiornamento, puoi ripristinare il AWS OpsWorks for Chef Automate server a Chef Automate 1 manualmente utilizzando il. AWS CLI

1. Eseguire il comando seguente per visualizzare il `BackupId` dell'ultimo backup eseguito sul server prima di tentare l'aggiornamento. Aggiungere il parametro `--region` se il server si trova in una regione AWS diversa da quella predefinita.

```
aws opsworks-cm describe-backups server_name
```

IDs I backup sono nel formato*ServerName-yyyyMMddHHmmssSSS*. Cercare nei risultati le seguenti proprietà di Chef Automate 1.

```
"Engine": "Chef"
"EngineVersion": "12.17.33"
```

2. Eseguire il comando seguente, utilizzando l'ID backup restituito nella fase 1 come valore di `--backup-id`.

```
aws opsworks-cm restore-server --server-name server_name --backup-id ServerName-yyyyMMddHHmmssSSS
```

Consentire tra 20 minuti e tre ore per il ripristino del server, a seconda della quantità di dati memorizzati sul server. Durante l'operazione di ripristino, il server è nello stato `RESTORING`. Questo stato viene visualizzato nella pagina delle proprietà del server in e restituito nei risultati del describe-servers comando. AWS Management Console

3. Al termine del ripristino, la console visualizza il messaggio Restore completed successfully (Ripristino completato). Il AWS OpsWorks for Chef Automate server è online e lo stesso di prima dell'avvio del processo di aggiornamento.

Vedi anche

- Manutenzione del sistema in AWS OpsWorks for Chef Automate
- Ripristino di un AWS OpsWorks for Chef Automate server da un backup
- DescribeServers nella Documentazione di riferimento API di AWS OpsWorks
- StartMaintenance nella Documentazione di riferimento API di AWS OpsWorks

Guida introduttiva con AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti

di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

AWS OpsWorks for Chef Automate ti consente di eseguire un server Chef Automate in. AWS Puoi effettuare il provisioning di un server Chef in circa 15 minuti.

A partire dal 3 maggio 2021, AWS OpsWorks for Chef Automate memorizza alcuni attributi del server Chef Automate in. AWS Secrets Manager Per ulteriori informazioni, consulta Integrazione con AWS Secrets Manager.

La seguente procedura dettagliata ti aiuta a creare il tuo primo server Chef in. AWS OpsWorks for Chef Automate

Prerequisiti

Prima di iniziare, è necessario completare i seguenti prerequisiti.

Argomenti

- Impostare una VPC
- Prerequisiti per l'utilizzo di un dominio personalizzato (opzionale)
- Configura una coppia di EC2 chiavi (opzionale)

Impostare una VPC

Il tuo AWS OpsWorks for Chef Automate server deve funzionare in un Amazon Virtual Private Cloud. Puoi aggiungerlo in una VPC, utilizzare la VPC predefinita, oppure creare una nuova VPC che contenga il server. Per informazioni su Amazon VPC e su come creare un nuovo VPC, consulta la Amazon VPC Getting Started Guide.

Se utilizzi una VPC esistente o ne crei una nuova, questa deve disporre delle seguenti impostazioni o proprietà.

- Il VPC deve avere almeno una sottorete.

  Se il tuo AWS OpsWorks for Chef Automate server sarà accessibile pubblicamente, rendi pubblica la sottorete e abilita l'assegnazione automatica di IP pubblici.
- La DNS resolution (Risoluzione DNS) deve essere abilitata.
- Nella sottorete, abilitare Auto-assign Public IP (Assegna automaticamente IP pubblico).

Prerequisiti per l'utilizzo di un dominio personalizzato (opzionale)

È possibile configurare il server Chef Automate nel proprio dominio, specificando un endpoint pubblico in un dominio personalizzato da utilizzare come endpoint del server. Quando si utilizza un dominio personalizzato, sono necessarie tutte le seguenti operazioni, come descritto nel dettaglio in questa sezione.

Argomenti

- Configurare un dominio personalizzato
- Ottenere un certificato.
- Ottenere una chiave privata

Configurare un dominio personalizzato

Per eseguire il server Chef Automate sul proprio dominio personalizzato, è necessario un endpoint pubblico di un server, ad esempio `https://aws.my-company.com`. Se si specifica un dominio personalizzato, è necessario fornire anche un certificato e una chiave privata, come descritto nelle sezioni precedenti.

Per accedere al server dopo la creazione, aggiungere un record DNS CNAME nel servizio DNS preferito. Questo record deve puntare il dominio personalizzato all'endpoint (il valore dell'attributo `Endpoint` del server) generato dal processo di creazione del server Chef Automate. Non è possibile accedere al server utilizzando il valore `Endpoint` generato se il server utilizza un dominio personalizzato.

Ottenere un certificato.

Per configurare il server Chef Automate sul proprio dominio personalizzato, è necessario un certificato HTTPS in formato PEM. Può essere un singolo certificato autofirmato o una catena di certificati. Quando si completa il flusso di lavoro Create Chef Automate server (Crea server Chef Automate), se si specifica questo certificato, è necessario fornire anche un dominio e una chiave privata personalizzati.

Di seguito sono riportati i requisiti per il valore del certificato:

- È possibile fornire un certificato personalizzato autofirmato o l'intera catena di certificati.
- Il certificato deve essere un certificato X509 valido o una catena di certificati in formato PEM.

- Il certificato deve essere valido al momento del caricamento. Non è possibile utilizzare un certificato prima dell'inizio del periodo di validità (la data `NotBefore` del certificato) o dopo la data di scadenza (la data `NotAfter` del certificato).
- Il nome comune del certificato o i nomi alternativi del soggetto (SANs), se presenti, devono corrispondere al valore del dominio personalizzato.
- Il certificato deve corrispondere al valore del campo Custom private key (Chiave privata personalizzata).

Ottenere una chiave privata

Per configurare il server Chef Automate nel proprio dominio personalizzato, è necessaria una chiave privata in formato PEM per la connessione al server tramite HTTPS. La chiave privata non deve essere crittografata; non può essere protetta da password o da passphrase. Se si specifica una chiave privata personalizzata, è necessario fornire anche un dominio e un certificato personalizzati.

Configura una coppia di EC2 chiavi (opzionale)

Ai fini della gestione del server Chef, non è necessaria né raccomandata una connessione SSH; puoi utilizzare i comandi `knife` per eseguire la maggior parte delle attività di gestione sul server Chef.

È necessaria una coppia di EC2 chiavi per connettersi al server utilizzando SSH nel caso in cui si perda o si desideri modificare la password di accesso per la dashboard di Chef Automate. Puoi utilizzare una coppia di chiavi esistente o crearne una nuova. Per ulteriori informazioni su come creare una nuova coppia di EC2 chiavi, consulta Amazon EC2 Key Pairs.

Se non hai bisogno di una EC2 key pair, sei pronto per creare un server Chef.

Creare un server Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

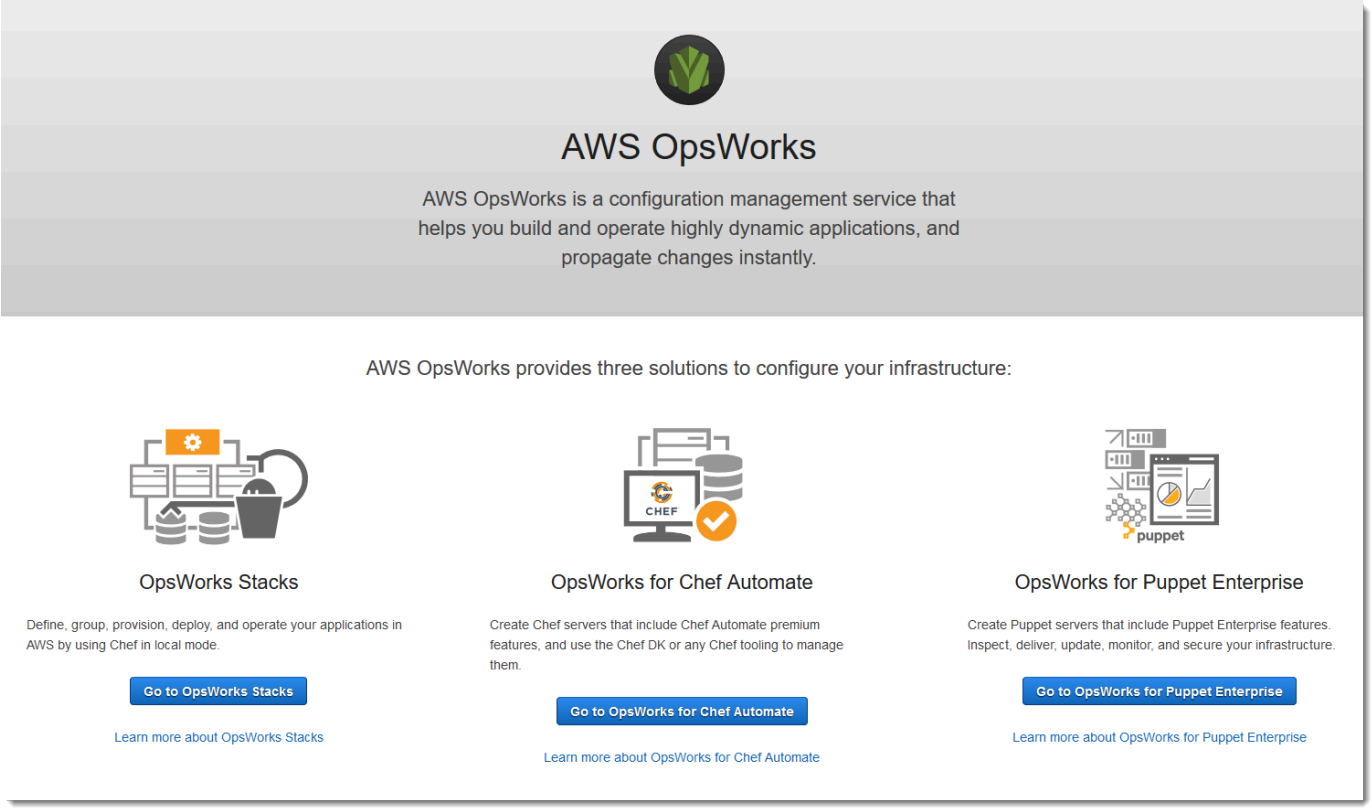
Puoi creare un server Chef utilizzando la AWS OpsWorks for Chef Automate console o il. AWS CLI

Argomenti

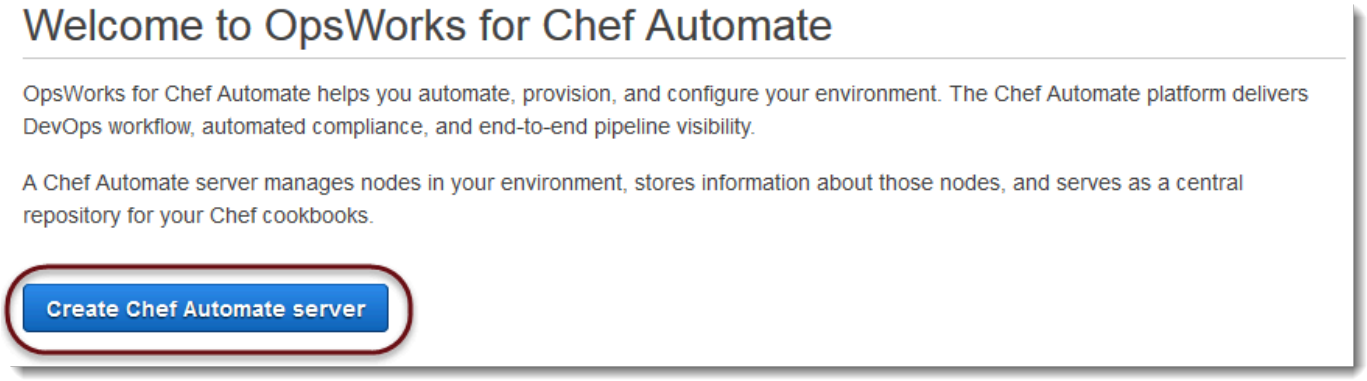
- Crea un server Chef Automate in AWS Management Console
- Crea un server Chef Automate utilizzando il AWS CLI

Crea un server Chef Automate in AWS Management Console

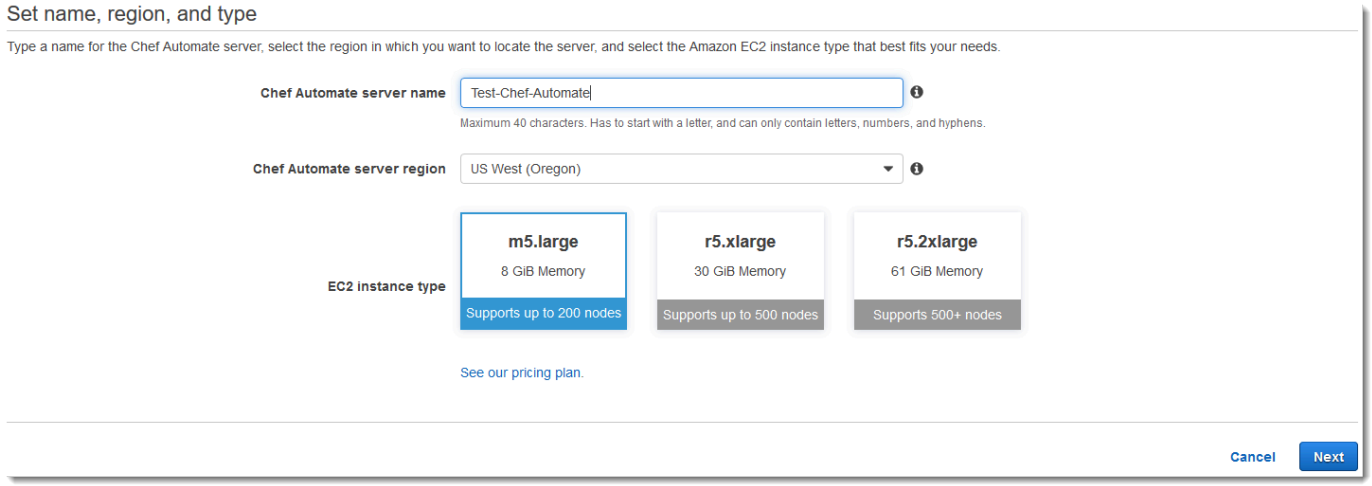
1. Accedi a AWS Management Console e apri la AWS OpsWorks console all'indirizzo https://console.aws.amazon.com/opsworks/.
2. Nella AWS OpsWorks home page, scegli Vai a OpsWorks Chef Automate.



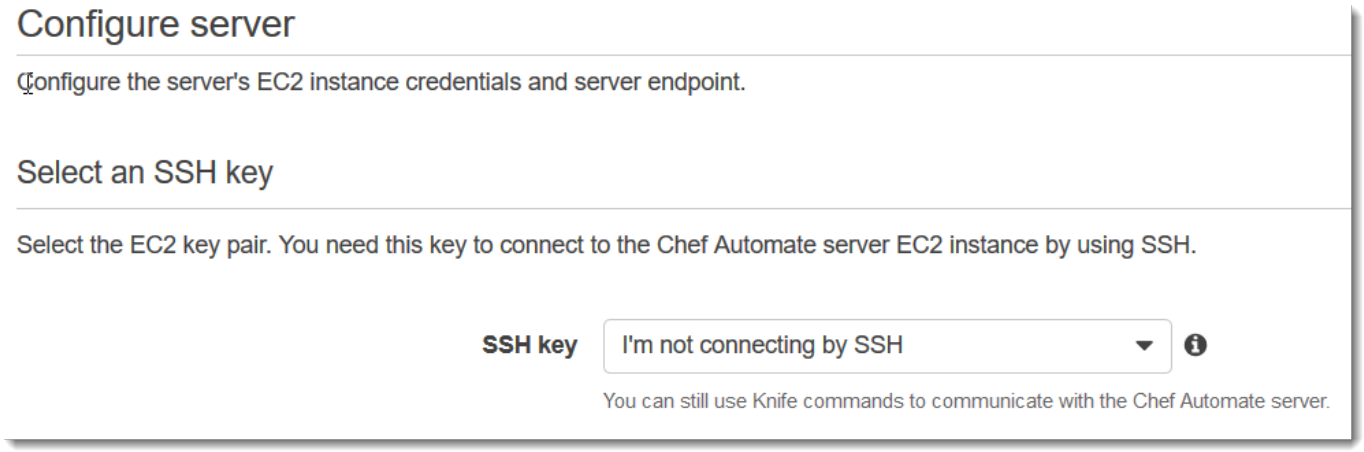
3. Nella AWS OpsWorks for Chef Automate home page, scegli Create Chef Automate server.



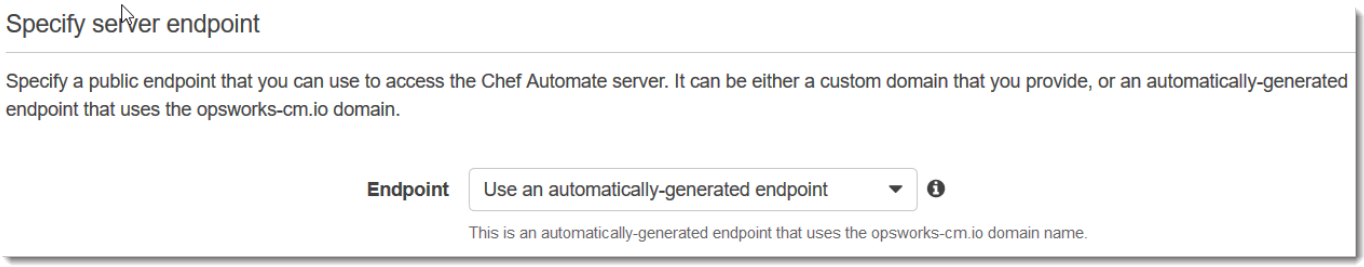
4. Nella pagina Set name, region, and type (Imposta nome, regione e tipo) specificare un nome per il server. I nomi del server Chef possono avere un massimo di 40 caratteri e devono contenere solo caratteri alfanumerici e trattini. Seleziona una regione supportata, quindi scegli un tipo di istanza che supporti il numero di nodi che desideri gestire. Se necessario, puoi modificare il tipo di istanza dopo la creazione del tuo server. Per questa procedura dettagliata, stiamo creando un tipo di istanza m5.large nella regione degli Stati Uniti occidentali (Oregon). Scegli Next (Successivo).



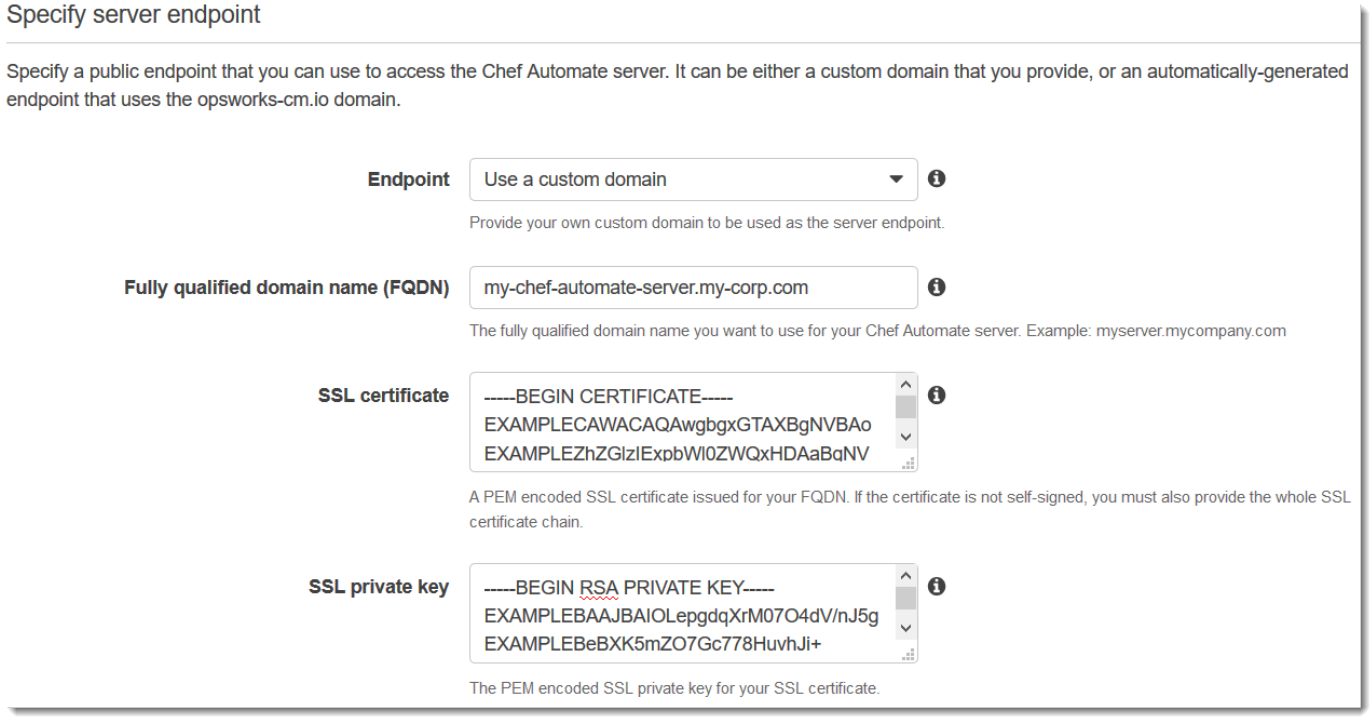
5. Nella pagina Configure server (Configura server), lasciare la selezione predefinita nell'elenco a discesa SSH key (Chiave SSH), a meno che non si desideri specificare il nome di una coppia di chiavi.



6. Per Specify server endpoint (Specifica endpoint del server), lasciare l'impostazione predefinita, Use an automatically-generated endpoint (Utilizza un endpoint generato automaticamente) e quindi scegliere Next (Avanti), a meno che non si desideri che il server sia in un dominio personalizzato di proprietà dell'utente. Per configurare un dominio personalizzato, passare alla fase successiva.



7. Per utilizzare un dominio personalizzato, per Specify server endpoint (Specifica endpoint del server), scegliere Use a custom domain (Usa un dominio personalizzato) dall'elenco a discesa.



a. Per Fully qualified domain name (FQDN) (Nome di dominio completo (FQDN)), specificare un nome completo (FQDN). È necessario essere proprietari del nome di dominio che si desidera utilizzare.

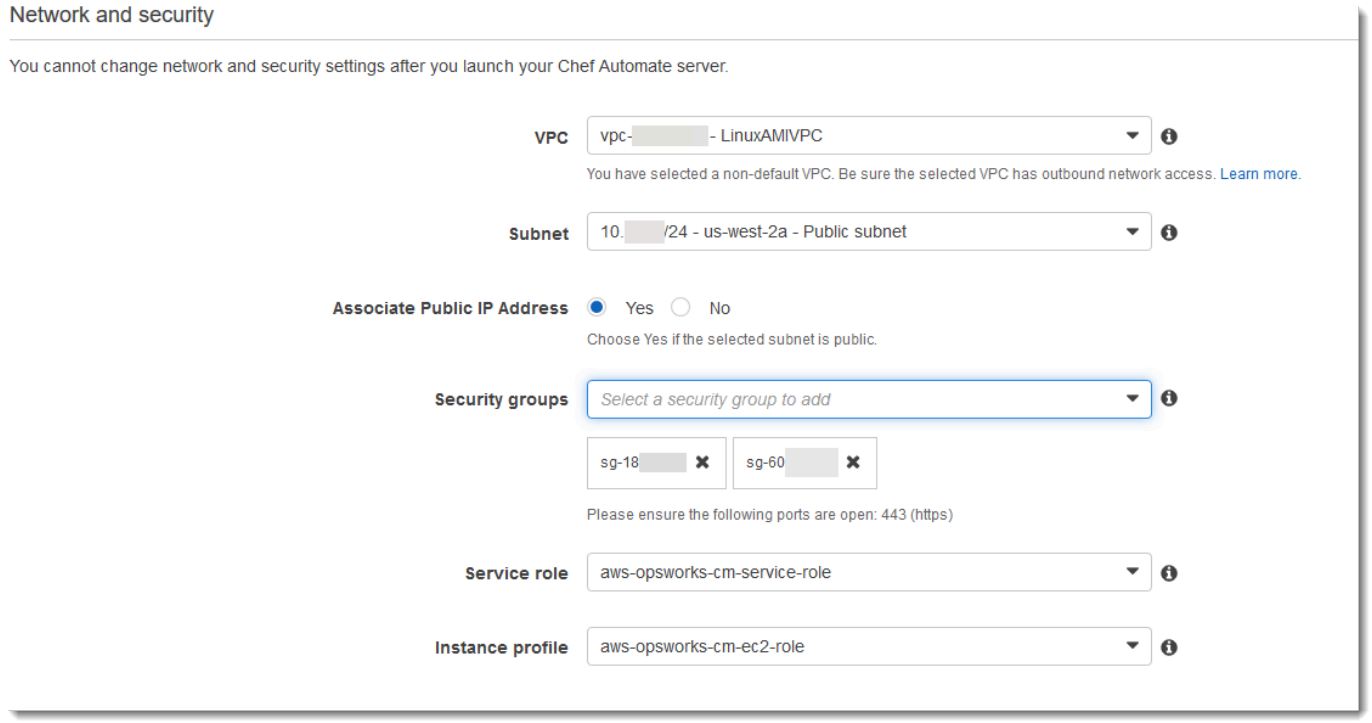
b. Per SSL certificate (Certificato SSL), incollare l'intero certificato in formato PEM, iniziando con `-----BEGIN CERTIFICATE-----` e terminando con `-----END CERTIFICATE-----`. L'oggetto del certificato SSL deve corrispondere al nome di dominio completo immesso nella fase precedente.

c. Per SSL private key (Chiave privata SSL), incollare l'intera chiave privata RSA, iniziando con `-----BEGIN RSA PRIVATE KEY-----` e finendo con `-----END RSA PRIVATE KEY-----`. La chiave privata SSL deve corrispondere alla chiave pubblica nel certificato SSL immesso nella fase precedente. Scegli Next (Successivo).

8. Nella pagina Configure Advanced Settings(Configura impostazioni avanzate) nell'area Network and Security (Rete e sicurezza), selezionare un VPC, una sottorete e uno o più gruppi di sicurezza. Di seguito sono riportati i requisiti del VPC:

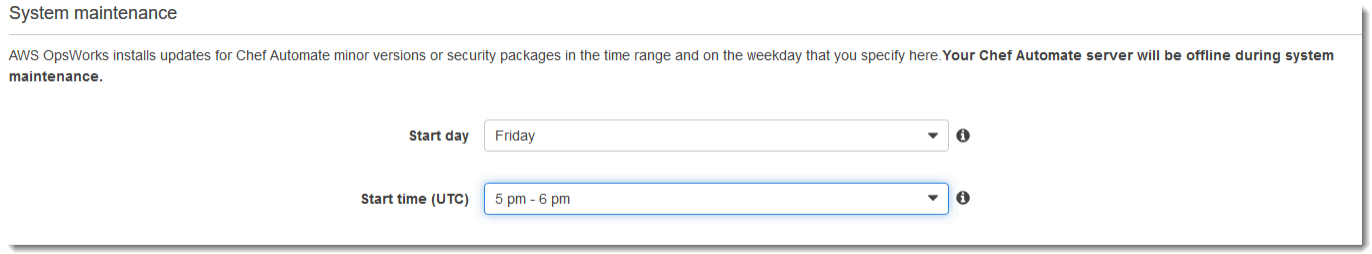
   - Il VPC deve avere almeno una sottorete pubblica.
   - La risoluzione DNS deve essere abilitata.
   - Auto-assign public IP (Assegnazione automatica dell'IP pubblico) deve essere abilitata nelle sottoreti pubbliche.

AWS OpsWorks può generare per te un gruppo di sicurezza, un ruolo di servizio e un profilo di istanza, se non ne hai già uno che desideri utilizzare. Il tuo server può essere un membro di più gruppi di sicurezza. Non puoi modificare le impostazioni di rete e di sicurezza per il server Chef dopo aver lasciato questa pagina.

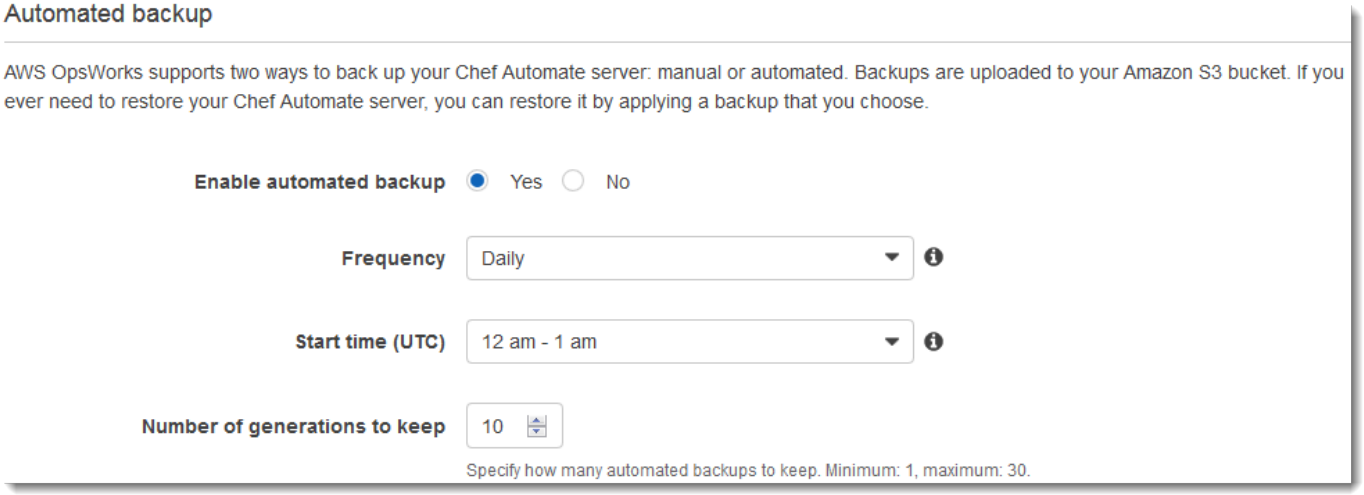


9. Nella sezione System maintenance (Manutenzione sistema), impostare il giorno e l'ora quando si desidera che il mantenimento del sistema inizi. In quanto ci si deve aspettare che il server sia offline durante il mantenimento del sistema, scegliere un orario di utilizzo del server basso durante l'orario di ufficio regolare. I nodi connessi inseriscono lo stato `pending-server` fino al completamento della manutenzione.

   La finestra di manutenzione è obbligatoria. È possibile modificare il giorno e l'ora di inizio in un secondo momento utilizzando AWS Management Console AWS CLI, o APIs.



10. Configurare i backup. I backup automatici sono abilitati per impostazione predefinita. Imposta una frequenza e un'ora preferite per l'avvio del backup automatico e imposta il numero di generazioni di backup da archiviare in Amazon Simple Storage Service. Vengono conservati un massimo di 30 backup; quando viene raggiunto il numero massimo, AWS OpsWorks for Chef Automate elimina i backup più vecchi per fare spazio a quelli nuovi.



11. (Facoltativo) In Tag, aggiungi tag al server e alle risorse correlate, come l' EC2 istanza, l'indirizzo IP elastico, il gruppo di sicurezza, il bucket S3 e i backup. Per ulteriori informazioni sull'assegnazione di tag a un AWS OpsWorks for Chef Automate server, consulta. Lavorare con i tag sulle AWS OpsWorks for Chef Automate risorse
12. Quando la configurazione delle impostazioni avanzate è terminata, selezionare Next (Successivo).
13. Nella pagina Review (Esamina), esaminare le proprie opzioni. Quando si è pronti per creare il server, selezionare Launch (Avvia).

    Mentre aspetti di AWS OpsWorks creare il tuo server Chef, vai Configura il server Chef utilizzando lo Starter Kit e scarica lo Starter Kit e le credenziali della dashboard di Chef Automate. Non attendere che il server sia online per scaricare questi elementi.

    Al termine della creazione del server, il tuo server Chef sarà disponibile nella home page di AWS OpsWorks for Chef Automate , con lo stato online. Dopo che il server è online, il pannello di controllo Chef Automate è disponibile nel dominio del server, per una URL nel formato seguente: `https://your_server_name-random.region.opsworks-cm.io`.

Crea un server Chef Automate utilizzando il AWS CLI

La creazione di un AWS OpsWorks for Chef Automate server eseguendo AWS CLI comandi è diversa dalla creazione di un server nella console. Nella console, AWS OpsWorks crea automaticamente un ruolo di servizio e un gruppo di sicurezza, se non si specificano quelli esistenti che si desidera utilizzare. In AWS CLI, AWS OpsWorks può creare automaticamente un gruppo di sicurezza se non lo si specifica, ma non crea automaticamente un ruolo di servizio; è necessario fornire un ruolo di servizio ARN come parte del `create-server` comando. Nella console, mentre AWS OpsWorks crei il tuo server Chef Automate, scarichi lo starter kit Chef Automate e le credenziali di accesso per la dashboard di Chef Automate. Poiché non è possibile eseguire questa operazione quando si crea un AWS OpsWorks for Chef Automate server utilizzando il AWS CLI, si utilizza un'utilità di elaborazione JSON per ottenere le credenziali di accesso e lo starter kit dai risultati del comando dopo che il nuovo server è online. `create-server` AWS OpsWorks for Chef Automate In alternativa, puoi generare una nuova serie di credenziali di accesso e un nuovo kit nella console, dopo che il nuovo server AWS OpsWorks for Chef Automate è online.

Se sul tuo computer locale non è già in esecuzione AWS CLI, scaricalo e installalo AWS CLI seguendo le istruzioni di installazione nella AWS Command Line Interface User Guide. Questa sezione non descrive tutti i parametri che è puoi utilizzare con il comando `create-server`. Per ulteriori informazioni sui parametri `create-server`, consulta `create-server` nella documentazione di riferimento delle AWS CLI .

1. Assicurati di completare i prerequisiti, soprattutto Impostare una VPC, o assicurati di disporre già di una VPC esistente che desideri utilizzare. Per creare il tuo server Chef Automate, hai bisogno dell'ID di una sottorete.
2. Eventualmente puoi generare una chiave pivotal di Chef con OpenSSLe salvare la chiave per un file semplice e sicuro sul computer locale. La chiave pivotal viene generata automaticamente come parte del processo di creazione di server, nel caso in cui non ne fornissi una nel comando `create-server`. Se desideri ignorare questo passaggio, puoi ottenere la chiave pivotal Chef Automate dai risultati del comando `create-server`. Se scegli di generare la chiave pivotal utilizzando i comandi seguenti, assicurati di includere il parametro `-pubout`, perché il valore della chiave pivotal Chef Automate è la metà pubblica della coppia di chiavi RSA. Per ulteriori informazioni, consulta la Fase 6.

   ```
   umask 077
   openssl genrsa -out "pivotal" 2048
   openssl rsa -in "pivotal" -pubout
   ```

3. Crea un ruolo di servizio e un profilo di istanza.
4. Trova e copia i ruoli ARNs di servizio nel tuo account.

```
aws iam list-roles --path-prefix "/service-role/" --no-paginate
```

Nei risultati del comando `list-roles`, cercare le voci ARN dei ruoli di servizio che sono simili a quanto segue. Prendi nota del ruolo di servizio ARNs. Per creare il server Chef Automate, hai bisogno di questi valori.

```
{
    "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "ec2.amazonaws.com"
                }
            }
        ]
    },
    "RoleId": "AROZZZZZZZZZZQG6R22HC",
    "CreateDate": "2018-01-05T20:42:20Z",
    "RoleName": "aws-opsworks-cm-ec2-role",
    "Path": "/service-role/",
    "Arn": "arn:aws:iam::000000000000:role/service-role/aws-opsworks-cm-ec2-role"
},
{
    "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "opsworks-cm.amazonaws.com"
                }
            }
        ]
    },
    "RoleId": "AROZZZZZZZZZZZZZZZ6QE",
```

```
    "CreateDate": "2018-01-05T20:42:20Z",
    "RoleName": "aws-opsworks-cm-service-role",
    "Path": "/service-role/",
    "Arn": "arn:aws:iam::000000000000:role/service-role/aws-opsworks-cm-service-
role"
}
```

5. Trova e copia i profili ARNs delle istanze nel tuo account.

```
aws iam list-instance-profiles --no-paginate
```

Nei risultati del comando `list-instance-profiles`, cerca le voci ARN dei profili delle istanze che sono simili a quanto segue. Prendi nota del profilo dell'istanza ARNs. Per creare il server Chef Automate, hai bisogno di questi valori.

```
{
    "Path": "/",
    "InstanceProfileName": "aws-opsworks-cm-ec2-role",
    "InstanceProfileId": "EXAMPLEDC6UR3LTUW7VHK",
    "Arn": "arn:aws:iam::123456789012:instance-profile/aws-opsworks-cm-ec2-role",
    "CreateDate": "2017-01-05T20:42:20Z",
    "Roles": [
        {
            "Path": "/service-role/",
            "RoleName": "aws-opsworks-cm-ec2-role",
            "RoleId": "EXAMPLEE4STNUQG6R22HC",
            "Arn": "arn:aws:iam::123456789012:role/service-role/aws-opsworks-cm-
ec2-role",
            "CreateDate": "2017-01-05T20:42:20Z",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            }
        }
```

```
    ]
},
```

6. Crea il AWS OpsWorks for Chef Automate server eseguendo il `create-server` comando.

   - Il valore `--engine` è `ChefAutomate`, `--engine-model` è `Single` e `--engine-version` è `12`.
   - Il nome del server deve essere univoco all'interno del tuo AWS account, all'interno di ogni regione. I nomi dei server devono iniziare con una lettera; quindi sono consentiti lettere, numeri o trattini (-) fino a un massimo di 40 caratteri.
   - Utilizza l'ARN del profilo dell'istanza e l'ARN del ruolo di servizio copiati nella Fase 4 e 5.
   - I tipi di istanza validi sono `m5.large`, `r5.xlarge` o `r5.2xlarge`. Per ulteriori informazioni sulle specifiche di questi tipi di istanze, consulta Instance Types nella Amazon EC2 User Guide.
   - Il parametro `--engine-attributes` è opzionale; se non specifichi uno o entrambi i valori, il processo di creazione del server genera i valori al posto tuo. Se aggiungi `--engine-attributes`, specifica il valore `CHEF_AUTOMATE_PIVOTAL_KEY` che hai generato nella Fase 2, un valore di `CHEF_AUTOMATE_ADMIN_PASSWORD` o entrambi.

     Se non imposti un valore per `CHEF_AUTOMATE_ADMIN_PASSWORD`, viene generata e restituita una password come parte della risposta `create-server`. Inoltre puoi scaricare nuovamente lo starter kit nella console, il quale rigenera questa password. La lunghezza della password è compresa tra 8 e 32 caratteri. La password può contenere lettere, numeri e caratteri speciali (`!/@#$%^+=_`). La password deve contenere almeno una lettera minuscola, una maiuscola, un numero e un carattere speciale.
   - Una coppia di chiavi SSH è facoltativa, ma può essere utile per connettersi al server Chef Automate se dovessi reimpostare la password amministratore del pannello di controllo di Chef Automate. Per ulteriori informazioni sulla creazione di una coppia di chiavi SSH, consulta Amazon EC2 Key Pairs nella Amazon EC2 User Guide.
   - Per utilizzare un dominio personalizzato, aggiungere i seguenti parametri al comando. Diversamente, il processo di creazione del server Chef Automate genera automaticamente un endpoint. Tutti e tre i parametri sono necessari per configurare un dominio personalizzato. Per informazioni sui requisiti aggiuntivi per l'utilizzo di questi parametri, consulta CreateServeril AWS OpsWorks CM API Reference.
     - `--custom-domain` - Un endpoint pubblico opzionale di un server, ad esempio `https://aws.my-company.com`.

  - `--custom-certificate` - Un certificato HTTPS in formato PEM. Il valore può essere un singolo certificato autofirmato o una catena di certificati.
  - `--custom-private-key` - Una chiave privata in formato PEM per la connessione al server tramite HTTPS. La chiave privata non deve essere crittografata; non può essere protetta da password o da passphrase.
- La manutenzione del sistema settimanale è obbligatoria. I valori validi devono essere specificati nel seguente formato: `DDD:HH:MM`. L'ora specificata è espressa nel Tempo coordinato universale (UTC). Se non si specifica un valore per `--preferred-maintenance-window`, il valore predefinito è un periodo casuale di un'ora il martedì, mercoledì o venerdì.
- I valori validi per `--preferred-backup-window` devono essere specificati in uno dei seguenti formati: `HH:MM` per backup giornaliero o `DDD:HH:MM` per backup settimanali. L'orario specificato è nel formato UTC. Il valore predefinito è un'ora di avvio giornaliera casuale. Per disattivare i backup automatici, aggiungere invece il parametro `--disable-automated-backup`.
- Per`--security-group-ids`, inserisci uno o più gruppi di sicurezza IDs, separati da uno spazio.
- Per `--subnet-ids`, inserire un ID di sottorete.

```
aws opsworks-cm create-server --engine "ChefAutomate" --engine-model "Single"
 --engine-version "12" --server-name "server_name" --instance-profile-arn
 "instance_profile_ARN" --instance-type "instance_type" --engine-attributes
 '{"CHEF_AUTOMATE_PIVOTAL_KEY":"pivotal_key","CHEF_AUTOMATE_ADMIN_PASSWORD":"password"}'
 --key-pair "key_pair_name" --preferred-maintenance-window
 "ddd:hh:mm" --preferred-backup-window "ddd:hh:mm" --security-group-
ids security_group_id1 security_group_id2 --service-role-arn "service_role_ARN" --
subnet-ids subnet_ID
```

Di seguito è riportato un esempio.

```
aws opsworks-cm create-server --engine "ChefAutomate" --engine-
model "Single" --engine-version "12" --server-name "automate-06" --
instance-profile-arn "arn:aws:iam::12345678912:instance-profile/aws-
opsworks-cm-ec2-role" --instance-type "m5.large" --engine-attributes
 '{"CHEF_AUTOMATE_PIVOTAL_KEY":"MZZE...Wobg","CHEF_AUTOMATE_ADMIN_PASSWORD":"zZZzDj2DLYXSZF
 --key-pair "amazon-test" --preferred-maintenance-window "Mon:08:00" --preferred-
backup-window "Sun:02:00" --security-group-ids sg-b00000001 sg-b0000008 --service-
```

```
role-arn "arn:aws:iam::12345678912:role/service-role/aws-opsworks-cm-service-role"
 --subnet-ids subnet-300aaa00
```

Nell'esempio seguente viene creato un server Chef Automate che utilizza un dominio personalizzato.

```
aws opsworks-cm create-server --engine "ChefAutomate" --engine-model "Single" --
engine-version "12" \
    --server-name "my-custom-domain-server" \
    --instance-profile-arn "arn:aws:iam::12345678912:instance-profile/aws-opsworks-
cm-ec2-role" \
    --instance-type "m5.large" \
    --engine-attributes
 '{"CHEF_AUTOMATE_PIVOTAL_KEY":"MZZE...Wobg","CHEF_AUTOMATE_ADMIN_PASSWORD":"zZZzDj2DLYXSZF
 \
    --custom-domain "my-chef-automate-server.my-corp.com" \
    --custom-certificate "-----BEGIN CERTIFICATE----- EXAMPLEqEXAMPLE== -----END
 CERTIFICATE-----" \
    --custom-private-key "-----BEGIN RSA PRIVATE KEY----- EXAMPLEqEXAMPLE= -----END
 RSA PRIVATE KEY-----" \
    --key-pair "amazon-test" \
    --preferred-maintenance-window "Mon:08:00" \
    --preferred-backup-window "Sun:02:00" \
    --security-group-ids sg-b00000001 sg-b0000008 \
    --service-role-arn "arn:aws:iam::12345678912:role/service-role/aws-opsworks-cm-
service-role" \
    --subnet-ids subnet-300aaa00
```

Nell'esempio seguente viene creato un server Chef Automate che aggiunge due tag: `Stage: Production` e `Department: Marketing`. Per ulteriori informazioni sull'aggiunta e la gestione dei tag sui AWS OpsWorks for Chef Automate server, Lavorare con i tag sulle AWS OpsWorks for Chef Automate risorse consulta questa guida.

```
aws opsworks-cm create-server --engine "ChefAutomate" --engine-model "Single" --
engine-version "12" \
    --server-name "my-test-chef-server" \
    --instance-profile-arn "arn:aws:iam::12345678912:instance-profile/aws-opsworks-
cm-ec2-role" \
    --instance-type "m5.large" \
    --engine-attributes
 '{"CHEF_AUTOMATE_PIVOTAL_KEY":"MZZE...Wobg","CHEF_AUTOMATE_ADMIN_PASSWORD":"zZZzDj2DLYXSZF
 \
```

```
    --key-pair "amazon-test" \
    --preferred-maintenance-window "Mon:08:00" \
    --preferred-backup-window "Sun:02:00" \
    --security-group-ids sg-b00000001 sg-b0000008 \
    --service-role-arn "arn:aws:iam::12345678912:role/service-role/aws-opsworks-cm-
service-role" \
    --subnet-ids subnet-300aaa00 \
    --tags [{\"Key\":\"Stage\",\"Value\":\"Production\"},{\"Key\":\"Department\",
\"Value\":\"Marketing\"}]
```

7. AWS OpsWorks for Chef Automate richiede circa 15 minuti per creare un nuovo server. Non annullare l'output del comando `create-server` né chiudere la sessione shell, poiché l'output può contenere informazioni importanti che non vengono visualizzate nuovamente. Per ottenere le password e lo starter kit dai risultati `create-server`, passare alla fase successiva.

   Se si utilizza un dominio personalizzato con il server, nell'output del comando `create-server`, copiare il valore dell'attributo `Endpoint`. Di seguito è riportato un esempio.

   ```
   "Endpoint": "automate-07-exampleexample.opsworks-cm.us-east-1.amazonaws.com"
   ```

8. Se hai scelto di AWS OpsWorks for Chef Automate generare automaticamente una chiave e una password, puoi estrarle in formati utilizzabili dai `create-server` risultati utilizzando un processore JSON come jq. Dopo aver installato jq, puoi eseguire i seguenti comandi per estrarre la chiave pivotal, la password amministratore del pannello di controllo di Chef Automate e lo starter kit. Se non hai fornito la chiave pivotal e la password nella Fase 4, assicurati di salvare in una posizione comoda e sicura la chiave pivotal e la password amministratore estratte.

   ```
   #Get the Chef password:
   cat resp.json | jq -r '.Server.EngineAttributes[] | select(.Name ==
    "CHEF_AUTOMATE_ADMIN_PASSWORD") | .Value'

   #Get the Chef Pivotal Key:
   cat resp.json | jq -r '.Server.EngineAttributes[] | select(.Name ==
    "CHEF_AUTOMATE_PIVOTAL_KEY") | .Value'

   #Get the Chef Starter Kit:
   cat resp.json | jq -r '.Server.EngineAttributes[] | select(.Name ==
    "CHEF_STARTER_KIT") | .Value' | base64 -D > starterkit.zip
   ```

9. Facoltativamente, se non avete estratto lo starter kit dai risultati dei `create-server` comandi, potete scaricare un nuovo starter kit dalla pagina Proprietà del server nella console. AWS

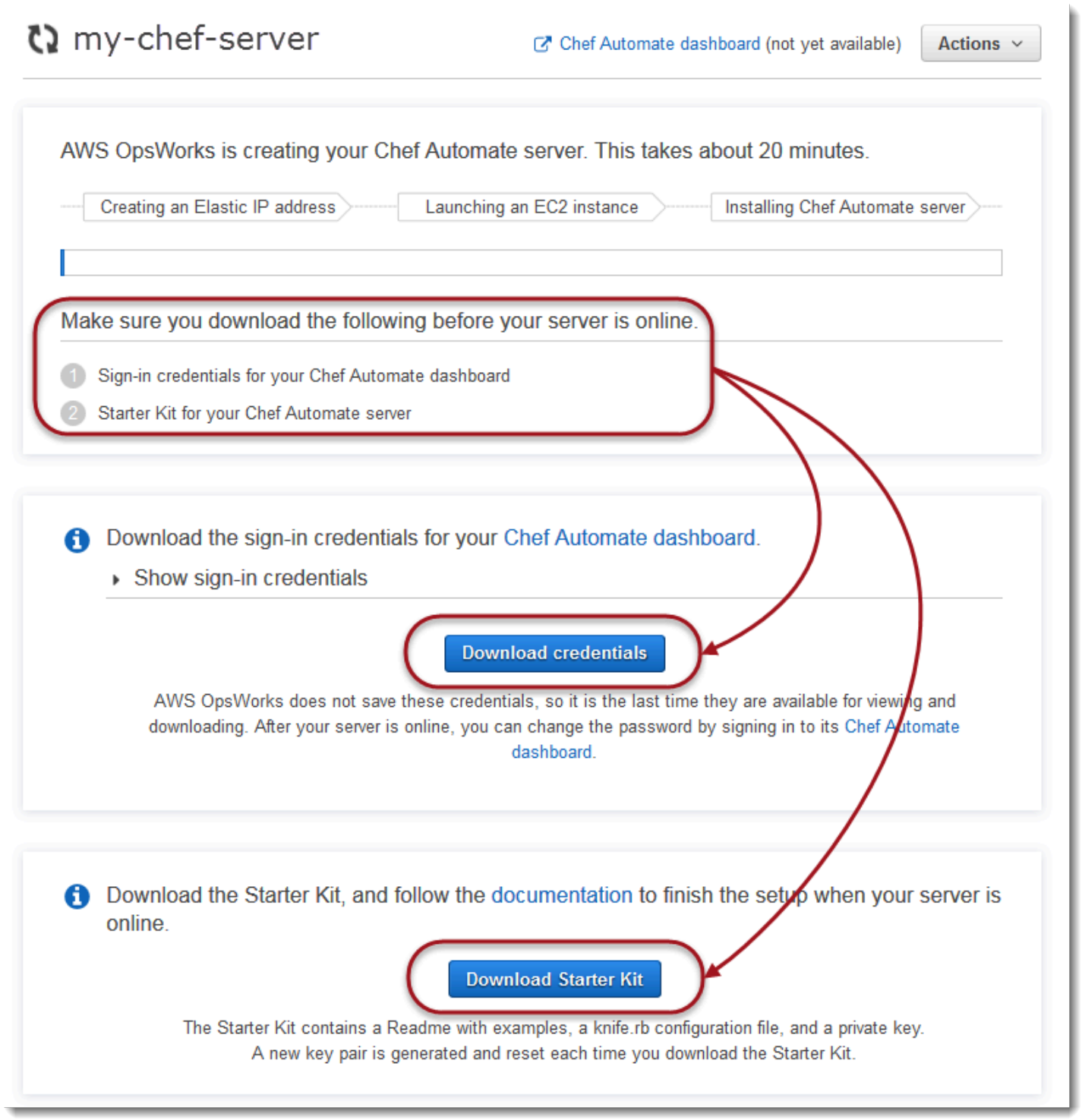
OpsWorks for Chef Automate Effettuare il download di un nuovo starter kit reimposta la password amministratore del pannello di controllo di Chef Automate.

10. Se non si utilizza un dominio personalizzato, passare alla fase successiva. Se utilizzate un dominio personalizzato con il server, create una voce CNAME nello strumento di gestione DNS aziendale per indirizzare il dominio personalizzato all' AWS OpsWorks for Chef Automate endpoint copiato nel passaggio 7. Non è possibile raggiungere o eseguire l'accesso al server con un dominio personalizzato fino a quando non si completa questa fase.
11. Al termine del processo di creazione del server, vai su the section called “Terminare la configurazione e caricare i libri di ricette”.

Configura il server Chef utilizzando lo Starter Kit

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Mentre la creazione del server Chef è ancora in corso, apri la relativa pagina Proprietà nella console AWS OpsWorks for Chef Automate . La prima volta che utilizzi un nuovo server Chef, la pagina Proprietà chiede di scaricare due elementi necessari. Scarica questi elementi prima che il server Chef sia online; i pulsanti del download non sono disponibili dopo che un nuovo server è online.



- Sign-in credentials for the Chef server. (Credenziali di accesso per il server Chef) Utilizzerai queste credenziali per accedere alla dashboard di Chef Automate, dove lavori con le funzionalità premium di Chef Automate, come il flusso di lavoro e le scansioni di conformità. AWS OpsWorks non salva queste credenziali; questa è l'ultima volta che sono disponibili per la visualizzazione e il download.

Se necessario, puoi modificare la password che viene fornita con queste credenziali dopo aver eseguito l'accesso.

- Starter Kit. Lo Starter Kit contiene un file README con esempi, un file di configurazione `knife.rb` e una chiave di accesso privata per l'utente primario o pivotal. Ogni volta che scarichi lo Starter Kit, viene generata una nuova coppia di chiavi (la vecchia chiave viene reimpostata).

Oltre alle credenziali che funzionano solo con il nuovo server, il file.zip dello Starter Kit include un semplice esempio di repository Chef che funziona con qualsiasi server. AWS OpsWorks for Chef Automate Nell'archivio di Chef, puoi archiviare libri di ricette, ruoli, file di configurazione e altri artefatti per la gestione dei nodi tramite Chef. Consigliamo di archiviare questo archivio in un sistema di controllo della versione (ad esempio Git) e di trattarlo come un codice sorgente. Per informazioni ed esempi che mostrano come impostare un archivio di Chef gestito su Git, consulta l'articolo relativo alle informazioni sull'archivio di Chef nella documentazione di Chef.

Prerequisiti

1. Durante la creazione del server, scarica le credenziali di accesso per il server Chef e salvale in una posizione comoda e sicura.
2. Scarica lo Starter Kit e decomprimi il suo file .zip nella directory del tuo WorkSpace. Non condividere la chiave privata dello Starter Kit. Se altri utenti gestiranno il server Chef, aggiungili in seguito come amministratori del pannello di controllo di Chef Automate.
3. Scarica e installa Chef Workstation (precedentemente noto come Chef Development Kit o Chef DK) sul computer che utilizzerai per gestire il server e i nodi Chef. L'`knife`utilità fa parte di Chef Workstation. Per istruzioni, consulta Installa Chef Workstation sul sito Web di Chef.

Esplorare i contenuti dello starter kit

Lo starter kit ha i seguenti contenuti.

- `cookbooks/` - Una directory per i libri di ricette creati. La `cookbooks/` cartella contiene il `opsworks-webserver` ricettario, un ricettario incartato che dipende dal ricettario del sito web di Chef `nginx` Supermarket. `Policyfile.rb`il valore predefinito è Chef supermarket come fonte secondaria se le dipendenze del ricettario non sono disponibili nella directory. `cookbooks/`
- `Policyfile.rb` - Un file di policy basato su Ruby che definisce i libri di ricette, le dipendenze e gli attributi che diventano la policy per i nodi.

- `userdata.sh` e `userdata.ps1` - Puoi utilizzare i file di dati dell'utente per associare i nodi automaticamente dopo l'avvio del server Chef Automate. `userdata.sh` viene usato per il bootstrap dei nodi basati su Linux e `userdata.ps1` per i nodi basati su Windows.
- `Berksfile` - Puoi usare questo file se preferisci utilizzare Berkshelf e i comandi `berks` per il caricamento dei libri di ricette e relative dipendenze. In questo scenario, utilizziamo `Policyfile.rb` e i comandi Chef per i libri di ricette, le dipendenze e gli attributi.
- `README.md`, un file basato su Markdown che descrive come utilizzare lo starter kit per configurare il server Chef Automate per la prima volta.
- `.chef` è una directory nascosta che contiene un file di configurazione knife (`knife.rb`) e un file di chiave di autenticazione segreta (.pem).
  - `.chef/knife.rb` - Un file di configurazione knife (`knife.rb`). Il `knife.rb`file è configurato in modo che le operazioni degli `knife`strumenti di Chef vengano eseguite sul server. AWS OpsWorks for Chef Automate
  - `.chef/ca_certs/opsworks-cm-ca-2020-root.pem` - Una chiave privata SSL firmata da un'autorità di certificazione (CA) fornita da AWS OpsWorks. Questa chiave consente al server di identificarsi all'agente client Chef Infra sui nodi gestiti dal server.

Configurazione dell'archivio di Chef

Un archivio di Chef contiene più directory. Ogni directory nello Starter Kit contiene un file README, il quale descrive lo scopo della directory e illustra come utilizzarla per gestire i tuoi sistemi con Chef. Sono disponibili due modi per installare i libri di ricette sul server Chef: esegui i comandi `knife` o un comando Chef per caricare un file di policy (`Policyfile.rb`) nel server che scarica e installa i libri di ricette specificati. In questo scenario vengono utilizzati i comandi Chef e `Policyfile.rb` per l'installazione dei libri di ricette sul server.

1. Crea una directory sul computer locale per l'archivio dei libri di ricetta, ad esempio `chef-repo`. Dopo aver aggiunto libri di cucina, ruoli e altri file a questo repository, ti consigliamo di caricarlo o archiviarlo in un sistema sicuro con versioni diverse, come CodeCommit Git o Amazon S3.
2. Nella directory `chef-repo` crea le directory seguenti:
   - `cookbooks/`- Memorizza libri di cucina.
   - `roles/` - Archivia i ruoli nei formati `.rb` o `.json`.
   - `environments/` - Archivia gli ambienti nei formati `.rb` o `.json`.

Utilizzare Policyfile.rb per ottenere i libri di ricette da un'origine remota

In questa sezione, modifichi `Policyfile.rb` per specificare i libri di ricette, quindi esegui un comando Chef per caricare il file sul server e installare i libri di ricette.

1. Visualizzare `Policyfile.rb` nello starter kit. Per impostazione predefinita, `Policyfile.rb` include il libro di ricette per il wrapper `opsworks-webserver` che dipende dal libro di ricette `nginx` disponibile nel sito Web Chef Supermarket. Il libro di ricette `nginx` consente di installare e configurare un server Web sui nodi gestiti. Viene anche specificato il libro di ricette `chef-client` richiesto, che consente di installare l'agente client Chef Infra sui nodi gestiti.

   `Policyfile.rb` indica anche il libro di ricette opzionale Chef Audit, che è possibile utilizzare per configurare le scansioni di conformità sui nodi. Per ulteriori informazioni sulla configurazione delle scansioni di conformità e sul recupero dei risultati di conformità per i nodi gestiti, consultare Scansioni di conformità in AWS OpsWorks for Chef Automate. Se non si desidera configurare le scansioni di conformità e l'audit in questo momento, eliminare `'audit'` dalla sezione `run_list` e non specificare gli attributi del libro di ricette `audit` alla fine del file.

   ```
   # Policyfile.rb - Describe how you want Chef to build your system.
   #
   # For more information about the Policyfile feature, visit

   # https://docs.chef.io/policyfile.html


   # A name that describes what the system you're building with Chef does.

   name 'opsworks-demo-webserver'

   # The cookbooks directory is the preferred source for external cookbooks

   default_source :chef_repo, "cookbooks/" do |s|

     s.preferred_for "nginx", "windows", "chef-client", "yum-epel", "seven_zip",
                     "build-essential", "mingw", "ohai", "audit", "logrotate", "cron"

   end
   # Alternative source
   default_source :supermarket
   ```

```
# run_list: chef-client runs these recipes in the order specified.

run_list  'chef-client',
          'opsworks-webserver',
          'audit'
# add 'ssh-hardening' to your runlist to fix compliance issues detected by the ssh-
baseline profile

# Specify a custom source for a single cookbook:

cookbook 'opsworks-webserver', path: 'cookbooks/opsworks-webserver'


# Policyfile defined attributes

# Define audit cookbook attributes
default["opsworks-demo"]["audit"]["reporter"] = "chef-server-automate"
default["opsworks-demo"]["audit"]["profiles"] = [
  {
    "name": "DevSec SSH Baseline",
    "compliance": "admin/ssh-baseline"
  }
]
```

Di seguito è riportato un esempio di `Policyfile.rb` senza il libro di ricette `audit` e gli attributi, se si desidera configurare solo i server Web `nginx` per il momento.

```
# Policyfile.rb - Describe how you want Chef to build your system.
#
# For more information on the Policyfile feature, visit
# https://docs.chef.io/policyfile.html

# A name that describes what the system you're building with Chef does.
name 'opsworks-demo-webserver'

# Where to find external cookbooks:
default_source :supermarket

# run_list: chef-client will run these recipes in the order specified.
run_list  'chef-client',
          'opsworks-webserver'

# Specify a custom source for a single cookbook:
```

```
cookbook 'opsworks-webserver', path: 'cookbooks/opsworks-webserver'
```

Se si modifica `Policyfile.rb`, assicurarsi di salvare il file.

2. Scaricare e installare i libri di ricette definiti in `Policyfile.rb`.

   ```
   chef install
   ```

   La versione di tutti i libri di ricette è indicata nel file `metadata.rb` del libro di ricette. A ogni modifica di un libro di ricette, è necessario aumentare la versione del libro di ricette all'interno del file `metadata.rb`.

3. Se si è scelto di configurare le scansioni di conformità e conservare le informazioni del libro di ricette `audit` nel file di policy, eseguire il push della policy `opsworks-demo` sul server.

   ```
   chef push opsworks-demo
   ```

4. Se è stato completata la fase 3, verificare l'installazione della policy. Esegui il comando seguente.

   ```
   chef show-policy
   ```

   Il risultato deve essere simile al seguente:

   ```
   opsworks-demo-webserver
   =======================
   * opsworks-demo:  ec0fe46314
   ```

5. È ora possibile aggiungere o eseguire il bootstrap dei nodi al server Chef Automate. È possibile automatizzare l'associazione di nodi completando la procedura in Aggiungi nodi automaticamente in AWS OpsWorks for Chef Automate o aggiungendo un nodo alla volta completando la procedura in Aggiungi nodi singolarmente.

(Alternativa) Utilizzare Berkshelf per ottenere libri di ricette da un'originen remota

Berkshelf è uno strumento per la gestione dei libri di ricette e delle loro dipendenze. Se preferisci utilizzare Berkshelf invece di `Policyfile.rb` per installare i libri di ricette nella memoria locale, utilizza la procedura presente in questa sezione anziché quella della sezione precedente. Puoi

specificare quali libri di ricette e quali versioni utilizzare con il tuo server Chef e caricarli. Lo starter kit contiene un file denominato `Berksfile` che è possibile utilizzare per elencare i libri di ricette.

1. Per iniziare, aggiungere il libro di ricette `chef-client` al Berksfile incluso. Il libro di ricette `chef-client` configura il software dell'agente client Chef Infra in ogni nodo che si connette al server Chef Automate. Per ulteriori informazioni su questo libro di ricette, consulta l'articolo relativo al libro di ricette del client Chef su Chef Supermarket.
2. Usando un editor di testo, aggiungere un altro libro di ricette al Berksfile che installa un'applicazione server Web, ad esempio il libro di ricette `apache2` che installa il server Web Apache. Il file Berksfile dovrebbe essere simile al seguente.

   ```
   source 'https://supermarket.chef.io'
   cookbook 'chef-client'
   cookbook 'apache2'
   ```

3. Scarica e installa il libro di ricette sul computer locale.

   ```
   berks install
   ```

4. Carica il libro di ricette sul server Chef.

   Su Linux, esegui quanto segue.

   ```
   SSL_CERT_FILE='.chef/ca_certs/opsworks-cm-ca-2020-root.pem' berks upload
   ```

   Su Windows, esegui il seguente comando Chef Workstation in una PowerShell sessione. Prima di eseguire il comando, assicurati di impostare la politica di esecuzione PowerShell su`RemoteSigned`. Aggiungi `chef shell-init` per rendere disponibili i comandi di utilità di Chef Workstation a PowerShell.

   ```
   $env:SSL_CERT_FILE="ca_certs\opsworks-cm-ca-2020-root.pem"
   chef shell-init berks upload
   Remove-Item Env:\SSL_CERT_FILE
   ```

5. Verifica di aver installato il libro di ricette, visualizzando un elenco di libri di ricette attualmente disponibili sul server Chef Automate. Per aggiungerlo, eseguire il comando `knife` seguente:

   Sei pronto per aggiungere nodi da gestire con il AWS OpsWorks for Chef Automate server.

```
knife cookbook list
```

(Opzionale) Configurare **`knife`** per l'utilizzo di un dominio personalizzato

Se il server Chef Automate utilizza un dominio personalizzato, potrebbe essere necessario aggiungere il certificato PEM della CA principale che ha firmato la catena di certificati del server o il certificato PEM del server se il certificato è autofirmato. `ca_certs`è una sottodirectory `chef/` che contiene le autorità di certificazione (CAs) ritenute affidabili dall'utilità Chef. `knife`

È possibile ignorare questa sezione se non si utilizza un dominio personalizzato o se il certificato personalizzato è firmato da un'autorità di certificazione (CA) root considerata attendibile dal sistema operativo in uso. In caso contrario, configurare `knife` per l'attendibilità del certificato SSL del server Chef Automate, come descritto nelle fasi seguenti.

1. Esegui il comando seguente.

   ```
   knife ssl check
   ```

   Se i risultati sono simili ai seguenti, ignorare il resto della procedura e passare a Aggiungere nodi da gestire con il server Chef.

   ```
   Connecting to host my-chef-automate-server.my-corp.com:443
             Successfully verified certificates from 'my-chef-automate-server.my-
   corp.com'
   ```

   Se viene visualizzato un messaggio di errore simile al seguente, passare alla fase successiva.

   ```
   Connecting to host my-chef-automate-server.my-corp.com:443
             ERROR: The SSL certificate of my-chef-automate-server.my-corp.com could
    not be verified.
             ...
   ```

2. Eseguire `knife ssl fetch` per l'attendibilità dei certificati del server AWS OpsWorks for Chef Automate . In alternativa, è possibile copiare manualmente il certificato CA root in formato PEM del server nella directory che è il valore di `trusted_certs_dir` nell'output di `knife ssl check`. Per impostazione predefinita, questa directory si trova in `.chef/ca_certs/` nello Starter Kit. L'output visualizzato dovrebbe essere simile al seguente:

```
WARNING: Certificates from my-chef-automate-server.my-corp.com will be fetched and
 placed in your trusted_cert
          directory (/Users/username/starterkit/.chef/../.chef/ca_certs).

          Knife has no means to verify these are the correct certificates. You
 should
          verify the authenticity of these certificates after downloading.

          Adding certificate for my-chef-automate-server in /Users/users/
starterkit/.chef/../.chef/ca_certs/servv-aqtswxu20swzkjgz.crt
          Adding certificate for MyCorp_Root_CA in /Users/users/
starterkit/.chef/../.chef/ca_certs/MyCorp_Root_CA.crt
```

3. Eseguire nuovamente `knife ssl check`. L'output visualizzato dovrebbe essere simile al seguente:

```
Connecting to host my-chef-automate-server.my-corp.com:443
          Successfully verified certificates from 'my-chef-automate-server.my-
corp.com'
```

Ora è possibile utilizzare `knife` con il server Chef Automate.

Aggiungere nodi da gestire con il server Chef

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

L'agente `chef-client` esegue ricette di Chef su computer fisici o virtuali, chiamati nodi, i quali sono associati con il server. Puoi connettere computer o istanze in locale al server Chef affinché li gestisca, purché i nodi stiano eseguendo sistemi operativi supportati. La registrazione dei nodi con il server Chef installa il software dell'agente `chef-client` su questi nodi.

Puoi utilizzare i seguenti metodi per aggiungere nodi:

- Aggiungi note singolarmente eseguendo un `knife` comando che aggiunge o avvia un' EC2 istanza in modo che il server Chef possa gestirla. Per ulteriori informazioni, consulta Aggiungi nodi singolarmente.
- Aggiungi nodi automaticamente utilizzando uno script per eseguire l'associazione automatica dei nodi con il server Chef. Il codice nello starter kit mostra come aggiungere nodi automaticamente tramite il metodo automatico. Per ulteriori informazioni, consultare Aggiungi nodi automaticamente in AWS OpsWorks for Chef Automate.

Argomenti

- Aggiungi nodi singolarmente
- Aggiungi nodi automaticamente in AWS OpsWorks for Chef Automate

Aggiungi nodi singolarmente

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Questa sezione descrive come eseguire un `knife` comando che aggiunge, o avvia, un' EC2 istanza in modo che il server Chef possa gestirla.

La versione minima supportata di `chef-client` sui nodi associati a un server AWS OpsWorks for Chef Automate è la versione 13.x. Consigliamo di eseguire la versione più recente e stabile`chef-client`.

Argomenti

- (Opzionale) Specificare l'URL dell'autorità di certificazione (CA) root del server Chef Automate
- Sistemi operativi supportati
- Aggiungere nodi con Knife

(Opzionale) Specificare l'URL dell'autorità di certificazione (CA) root del server Chef Automate

Se il server utilizza un dominio e un certificato personalizzati, potrebbe essere necessario modificare la variabile `ROOT_CA_URL` nello script userdata con un URL pubblico che è possibile utilizzare per ottenere l'autorità di certificazione (CA) root in formato PEM del server. I seguenti AWS CLI comandi caricano la tua CA root in un bucket Amazon S3 e generano un URL predefinito che puoi utilizzare per un'ora.

1. Caricare il certificato CA root in formato PEM su S3.

   ```
   aws s3 cp ROOT_CA_PEM_FILE_PATH s3://bucket_name/
   ```

2. Generare un URL prefirmato che è possibile utilizzare per un'ora (3600 secondi, in questo esempio) per scaricare l'autorità di certificazione (CA) root.

   ```
   aws s3 presign s3://bucket_name/ROOT_CA_PEM_FILE_NAME --expires-in 3600
   ```

3. Modificare la variabile `ROOT_CA_URL` nello script userdata con il valore dell'URL pre-firmato.

Sistemi operativi supportati

Per l'elenco aggiornato dei sistemi operativi supportati per i nodi, consulta l'argomento sul sito Web di Chef.

Aggiungere nodi con Knife

Il `knife-ec2`plug-in è incluso in Chef Workstation. Se lo conosci meglio`knife-ec2`, puoi usarlo invece di effettuare il provisioning e `knife bootstrap` avviarne di nuovi. EC2instances Altrimenti, avvia una nuova EC2 istanza e segui i passaggi descritti in questa sezione.

Per aggiungere nodi da gestire

1. Eseguire il seguente comando `knife bootstrap`. Questo comando avvia un' EC2 istanza nei nodi che il server Chef gestirà. Nota che stai indicando al server Chef di eseguire le ricette dal libro di ricette `nginx` installato su the section called “Utilizzare Policyfile.rb per ottenere i libri di ricette da un'origine remota”. Per ulteriori informazioni sull'aggiunta di nodi mediante l'esecuzione del comando `knife bootstrap`, consulta come effettuare il bootstrap su un nodo nella documentazione di Chef.

La tabella riportata di seguito mostra i nomi utente validi per i sistemi operativi del nodo sul comando `knife` di questa fase. Se non funziona `root` né `ec2-user`, contatta il provider della tua AMI. Per ulteriori informazioni su come connettere le istanze basate su Linux, consulta l'argomento relativo alla connessione di istanze di Linux tramite SSH nella documentazione di AWS.

Valori validi per i nomi utente nei sistemi operativi del nodo

Sistema operativo	Nomi utente validi
Amazon Linux	`ec2-user`
Red Hat Enterprise Linux 5	`root` o `ec2-user`
Ubuntu	`ubuntu`
Fedora	`fedora` o `ec2-user`
SUSE Linux	`root` o `ec2-user`

```
knife bootstrap INSTANCE_IP_ADDRESS -N INSTANCE_NAME -x USER_NAME --sudo --run-list
 "recipe[nginx]"
```

2. Verifica che il nuovo nodo sia stato aggiunto eseguendo i seguenti comandi, sostituendoli *INSTANCE_NAME* con il nome dell'istanza che hai appena aggiunto.

```
knife client show INSTANCE_NAME
knife node show INSTANCE_NAME
```

Aggiungi nodi automaticamente in AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Questo argomento descrive come aggiungere automaticamente nodi Amazon Elastic Compute Cloud (Amazon EC2) al server Chef. Il codice nello starter kit mostra come aggiungere nodi automaticamente tramite il metodo automatico. Il metodo consigliato per l'associazione automatica di nuovi nodi, riguarda la configurazione di un libro di ricette del client Chef. Puoi utilizzare lo script `userdata` nello starter kit e modificare la sezione `run_list` dello script `userdata` o il file `Policyfile.rb` con i libri di ricette da applicare ai nodi. Prima di eseguire l'agente `chef-client` installa il libro di ricette del client Chef sul tuo server Chef, quindi installa l'agente `chef-client` in modalità servizio, ad esempio con un ruolo HTTPD, come mostrato nel comando esemplificativo di seguito.

```
chef-client -r "chef-client,role[httpd]"
```

Per comunicare con il server Chef, il software dell'agente `chef-client` deve poter accedere alla chiave pubblica del nodo del client. Puoi generare una coppia di chiavi pubblica-privata in Amazon e EC2 quindi passare la chiave pubblica alla chiamata AWS OpsWorks `associate-node` API con il nome del nodo. Lo script incluso nello starter kit, raccoglie per te il nome dell'organizzazione, il nome del server e l'endpoint del server. In questo modo, viene garantito che il nodo sia associato al server di Chef e che il software dell'agente `chef-client` in esecuzione sul nodo, possa comunicare con il server dopo aver abbinato la chiave privata.

La versione minima supportata di `chef-client` sui nodi associati a un server AWS OpsWorks for Chef Automate è la versione 13.x. Ti consigliamo di eseguire la versione più recente e stabile`chef-client`.

Per informazioni su come dissociare un nodo, Dissociare un nodo da un server AWS OpsWorks for Chef Automate consulta questa guida e `disassociate-node`la documentazione dell' AWS OpsWorks for Chef Automate API.

Argomenti

- Sistemi operativi supportati
- Passaggio 1: crea un ruolo IAM da utilizzare come profilo di istanza
- Fase 2: Installare il libro di ricette del client Chef
- Fase 3. Creare istanze tramite uno script di associazione automatica
- Altri metodi di esecuzioni ripetute in automatico relativi a chef-client
- Argomenti correlati

Sistemi operativi supportati

Per l'elenco aggiornato dei sistemi operativi supportati per i nodi, consulta l'argomento sul sito Web di Chef.

Passaggio 1: crea un ruolo IAM da utilizzare come profilo di istanza

Crea un ruolo AWS Identity and Access Management (IAM) da utilizzare come profilo di EC2 istanza e collega la seguente policy al ruolo IAM. Questa policy consente all'API AWS OpsWorks for Chef Automate (`opsworks-cm`) di comunicare con l' EC2 istanza durante la registrazione del nodo. Per ulteriori informazioni sui profili di istanza, consulta Using Instance Profiles nella EC2 documentazione di Amazon. Per informazioni su come creare un ruolo IAM, consulta Creazione di un ruolo IAM nella console nella EC2 documentazione di Amazon.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "opsworks-cm:AssociateNode",
                "opsworks-cm:DescribeNodeAssociationStatus",
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Fase 2: Installare il libro di ricette del client Chef

In caso contrario, segui le istruzioni su (Alternativa) Utilizzare Berkshelf per ottenere libri di ricette da un'originen remota per essere sicuro che il tuo Berksfile o il file `Policyfile.rb` faccia riferimento al libro di ricette del client Chef e che installi il libro di ricette.

Fase 3. Creare istanze tramite uno script di associazione automatica

1. Per creare EC2 istanze, puoi copiare `userdata` lo script dallo Starter Kit `userdata` nella sezione delle istruzioni dell' EC2 istanza, nelle configurazioni di avvio del gruppo Amazon EC2 Auto Scaling o in un modello. AWS CloudFormation Per ulteriori informazioni sull'aggiunta di script ai dati utente, consulta Running Commands on Your Linux Instance at Launch nella EC2 documentazione di Amazon.

Questo script esegue il `associate-node`comando `opsworks-cm` API per associare un nuovo nodo al server Chef.

Per impostazione predefinita, il nome del nuovo nodo registrato è l'ID dell'istanza, ma puoi cambiare il nome modificando il valore della variabile `NODE_NAME` nello script `userdata`. Poiché al momento non è possibile modificare il nome dell'organizzazione nell'interfaccia utente della console di Chef, lascia il valore `CHEF_AUTOMATE_ORGANIZATION` impostato su `default`.

2. Segui la procedura descritta in Avvio di un'istanza nella EC2 documentazione, con le modifiche apportate qui. Nella procedura guidata di avvio dell' EC2 istanza, scegli un'AMI Amazon Linux.
3. Sulla pagine Configure Instance Details (Configurazione dei dettagli dell'istanza), seleziona il ruolo creato su Passaggio 1: crea un ruolo IAM da utilizzare come profilo di istanza come ruolo IAM.
4. Nell'area Advanced Details (Dettagli avanzati), carica lo script `userdata.sh` creato precedentemente in questa procedura.
5. Non è necessario apportare alcuna modifica nella pagina Add Storage (Aggiungi storage). Vai su Add Tags (Aggiungi tag).
6. Nella pagina Configure Security Group (Configura gruppo di sicurezza), scegli Add Rule (Aggiungi regola), quindi scegli il tipo di HTTP per aprire le porte numero 443 e 80 per il server Web Apache in questo esempio.
7. Scegli Review and Launch (Analizza e lancia), quindi scegli Launch (Lancia). All'avvio del nuovo nodo, quest'ultimo applica le configurazioni specificate dalle ricette che hai specificato nel parametro `RUN_LIST`.
8. Facoltativo: se hai aggiunto il libro di ricette `nginx` al tuo elenco di esecuzione, quando aprirai la pagina Web collegata al DNS pubblico del tuo nuovo nodo, dovresti visualizzare un sito Web ospitato dal server Web nginx.

Altri metodi di esecuzioni ripetute in automatico relativi a **`chef-client`**

Sebbene sia più difficile da realizzare e non consigliato, puoi eseguire lo script in questo argomento esclusivamente come parte dei dati utente dell'istanza autonoma, utilizzare un AWS CloudFormation modello per aggiungerlo ai dati utente della nuova istanza, configurare un `cron` processo per eseguire lo script regolarmente o eseguirlo all'`chef-client`interno di un servizio. Tuttavia, consigliamo il metodo del libro di ricette del client Chef, visto che altre tecniche di automazione hanno diversi svantaggi.

Per un elenco completo dei parametri che puoi fornire a `chef-client`, consulta l'articolo relativo alla documentazione di Chef.

Argomenti correlati

I seguenti post del AWS blog offrono ulteriori informazioni sull'associazione automatica dei nodi al server Chef Automate, utilizzando i gruppi di Auto Scaling o all'interno di più account.

- Utilizzo di AWS OpsWorks for Chef Automate per gestire EC2 istanze con Auto Scaling
- OpsWorks per Chef Automate: avvio automatico dei nodi in diversi account

Accedere al pannello di controllo di Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Dopo aver scaricato le credenziali di accesso dalla pagina delle Proprietà del server Chef e il server è online, accedi al pannello di controllo Chef Automate. In questa procedura guidata, abbiamo consigliato di caricare prima i libri di ricette e di aggiungere almeno un nodo da gestire. In questo modo puoi visualizzare le informazioni sui libri di ricette e i nodi nel pannello di controllo.

Quando tenti di connetterti alla pagina web della dashboard, nel tuo browser vengono visualizzati avvisi relativi ai certificati finché non installi un certificato SSL AWS OpsWorks specifico e firmato da un'autorità di certificazione sul computer client che stai utilizzando per gestire il server Chef. Se non desideri visualizzare gli avvisi prima di continuare a navigare sulla pagina Web del pannello di controllo, installa il certificato SSL prima di effettuare l'accesso.

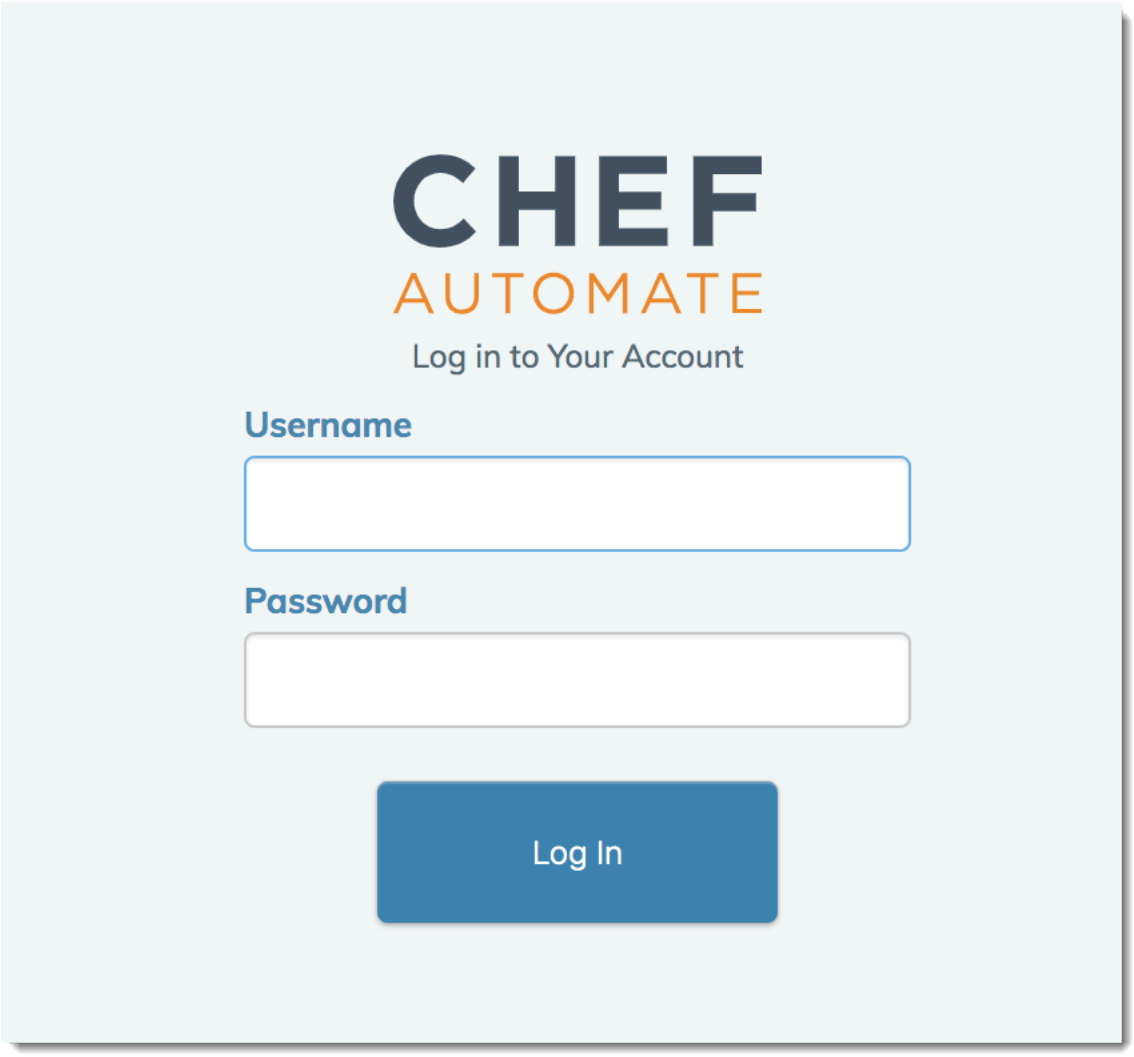
Dopo aver installato il certificato SSL lato client, puoi accedere al pannello di controllo di Chef Automate senza visualizzare i messaggi di avviso.

> **Note**
>
> Gli utenti di Google Chrome sui sistemi operativi Ubuntu e Linux Mint possono riscontrare dei problemi all'accesso. Consigliamo di utilizzare Mozilla Firefox o altri browser per accedere

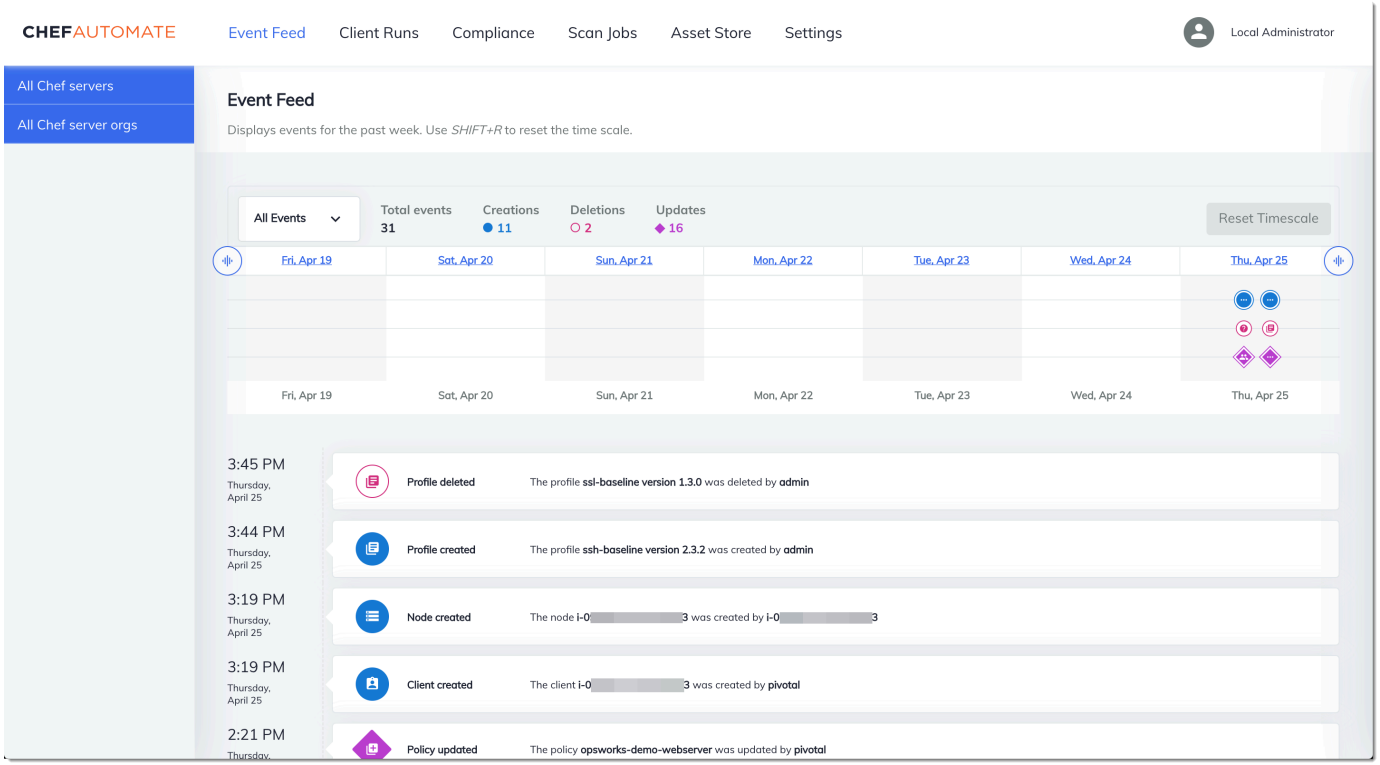
e utilizzare il pannello di controllo di Chef Automate su questi sistemi operativi. Non è stato riscontrato alcun problema utilizzando Google Chrome su Windows o Mac OS.

Per accedere al pannello di controllo di Chef Automate

1. Decomprimi e apri le credenziali di Chef Automate scaricate tramite Prerequisiti. Devi disporre di queste credenziali per poter accedere.
2. Apri la pagina Properties (Proprietà) sul tuo server Chef.
3. In alto a destra della pagina Properties (Proprietà), scegli Open Chef Automate dashboard (Apri pannello di controllo Chef Automate).
4. Accedi utilizzando le credenziali della Fase 1.



5. Sul pannello di controllo di Chef Automate dashboard, puoi visualizzare informazioni dettagliate dei nodi su cui hai eseguito il bootstrap, su eventi e progressi dell'esecuzione del libro di ricette, sul livello di conformità dei nodi e molto altro. Per ulteriori informazioni sulle caratteristiche del pannello di controllo di Chef Automate e su come utilizzarle, consulta la documentazione di Chef Automate.



> **Note**
>
> Per informazioni su come modificare la password utilizzata per accedere al pannello di controllo di Chef Automate, consulta Reimpostazione delle credenziali del pannello di controllo di Chef Automate.

Creare un AWS OpsWorks for Chef Automate server utilizzando AWS CloudFormation

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

AWS OpsWorks for Chef Automate ti consente di eseguire un server Chef Automate in. AWS Puoi effettuare il provisioning di un server Chef Automate in circa 15 minuti.

A partire dal 3 maggio 2021, AWS OpsWorks for Chef Automate memorizza alcuni attributi del server Chef Automate in. AWS Secrets Manager Per ulteriori informazioni, consulta Integrazione con AWS Secrets Manager.

La seguente procedura dettagliata ti aiuta a creare un server AWS OpsWorks for Chef Automate creando uno stack in. AWS CloudFormation

Argomenti

- Prerequisiti
- Creazione di un server Chef Automate in AWS CloudFormation

Prerequisiti

Prima di creare un nuovo server Chef Automate, è necessario creare le necessarie risorse al di fuori di AWS OpsWorks for Chef Automate per accedere e gestire il server Chef. Per ulteriori informazioni, consulta Prerequisiti nella sezione delle nozioni di base di questa guida.

Consulta la sezione OpsWorks -CM della Guida per l'AWS CloudFormation utente di riferimento per conoscere i valori supportati e obbligatori nel AWS CloudFormation modello utilizzato per creare il server.

Se si sta creando un server che utilizza un dominio personalizzato, è necessario un dominio, un certificato e una chiave privata personalizzati. È necessario specificare i valori per tutti e tre questi parametri nel AWS CloudFormation modello. Per ulteriori informazioni sui

requisiti per i `CustomPrivateKey` parametri, e`CustomDomain`,,`CustomCertificate`,,,,,,, CreateServerconsulta il AWS OpsWorks CM API Reference.

Crea un valore di password per l'attributo del motore `CHEF_AUTOMATE_ADMIN_PASSWORD`. La lunghezza della password è compresa tra 8 e 32 caratteri. La password può contenere lettere, numeri e caratteri speciali (`!/@#$%^+=_`). La password deve contenere almeno una lettera minuscola, una maiuscola, un numero e un carattere speciale. Specificate questa password nel AWS CloudFormation modello o come valore del `CHEF_AUTOMATE_ADMIN_PASSWORD` parametro durante la creazione dello stack.

Genera una coppia di chiavi RSA con codifica base64 prima di iniziare a creare un server Chef Automate in. AWS CloudFormation La chiave pubblica della coppia è il valore di, lo Chef`CHEF_AUTOMATE_PIVOTAL_KEY`, specifico dell'API. EngineAttributesCreateServer Questa chiave viene fornita come valore di Parameters nella AWS CloudFormation console o nel create-stack comando in AWS CLI. Per generare questa chiave, suggeriamo i seguenti metodi.

- Nei computer basati su Linux, puoi generare questa chiave eseguendo il comando OpenSSL.

  ```
  openssl genrsa -out pivotal_key_file_name.pem 2048
  ```

  Quindi esporta la porzione della chiave pubblica RSA della coppia in un file. La chiave pubblica diventa il valore di `CHEF_AUTOMATE_PIVOTAL_KEY`.

  ```
  openssl rsa -in pivotal_key_file_name.pem -pubout -out public.pem -outform PEM
  ```

- Sui computer basati su Windows, è possibile utilizzare l'TTYgen utilità Pu per generare una coppia di chiavi RSA con codifica base64. Per ulteriori informazioni, vedere Pu TTYgen - Key Generator for PuTTY su Windows su SSH.com.

Creazione di un server Chef Automate in AWS CloudFormation

Questa sezione descrive come utilizzare un AWS CloudFormation modello per creare uno stack che crea un server. AWS OpsWorks for Chef Automate È possibile farlo utilizzando la AWS CloudFormation console o il AWS CLI. È disponibile un AWS CloudFormation modello di esempio da utilizzare per creare uno stack di AWS OpsWorks for Chef Automate server. Assicurati di aggiornare il modello di esempio con il nome del server, i ruoli IAM, il profilo dell'istanza, la descrizione del server, il numero di backup, le opzioni di manutenzione e i tag opzionali. Se il server utilizzerà un dominio personalizzato, è necessario specificare i valori per `CustomDomainCustomCertificate`,

e `CustomPrivateKey` i parametri nel AWS CloudFormation modello. È possibile specificare gli attributi `CHEF_AUTOMATE_PIVOTAL_KEY` del motore `CHEF_AUTOMATE_ADMIN_PASSWORD` e i relativi valori nel AWS CloudFormation modello oppure fornire solo gli attributi e quindi specificare i valori per gli attributi nella procedura guidata o create-stack nel AWS CloudFormation comando Create Stack. Per ulteriori informazioni su tali attributi, consulta the section called “Crea un server Chef Automate in AWS Management Console” nella sezione delle nozioni di base di questa guida.

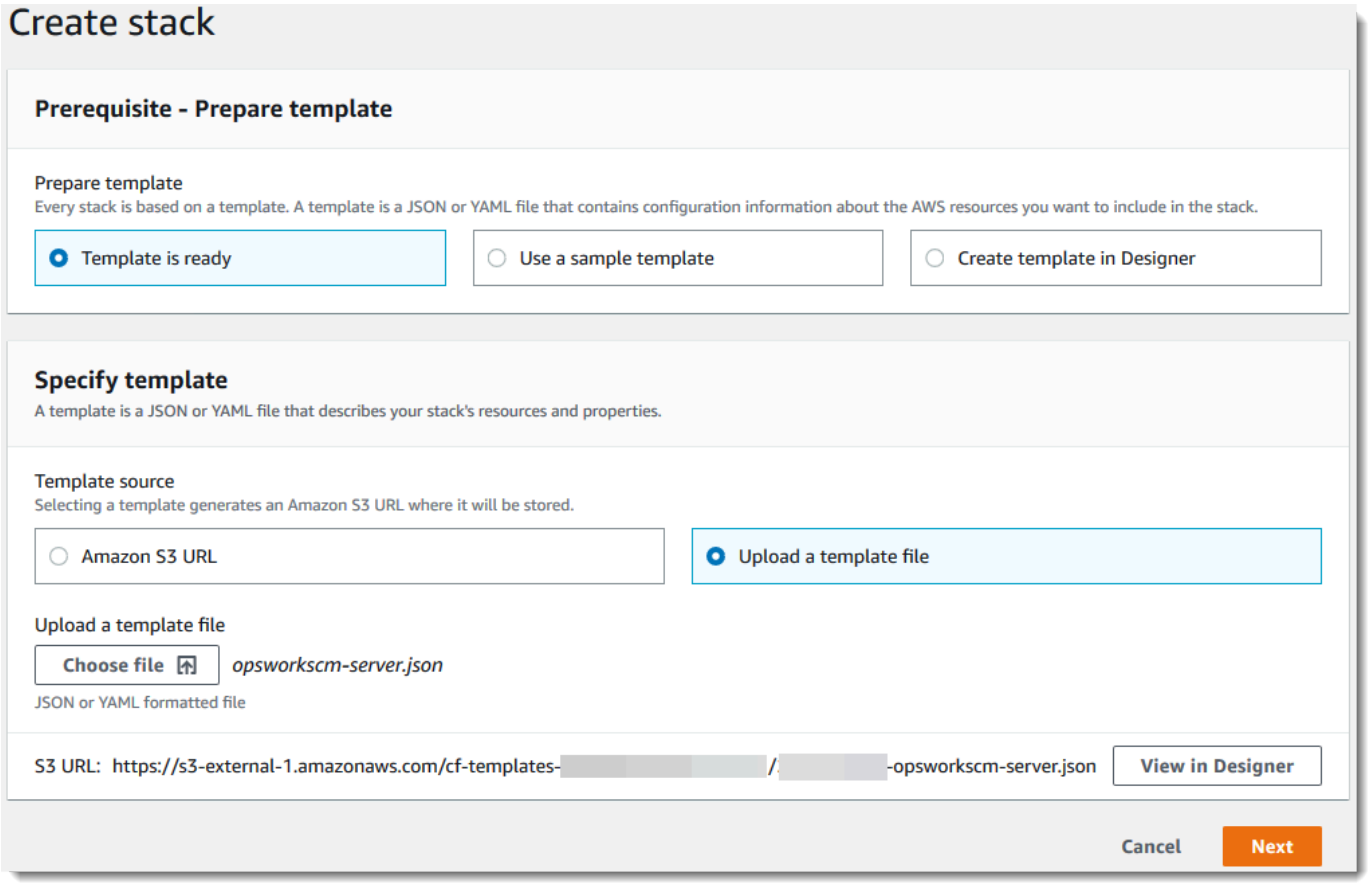
Argomenti

- Creazione di un server Chef Automate tramite la AWS CloudFormation (Console)
- Creazione di un server Chef Automate tramite la CLI AWS CloudFormation

Creazione di un server Chef Automate tramite la AWS CloudFormation (Console)

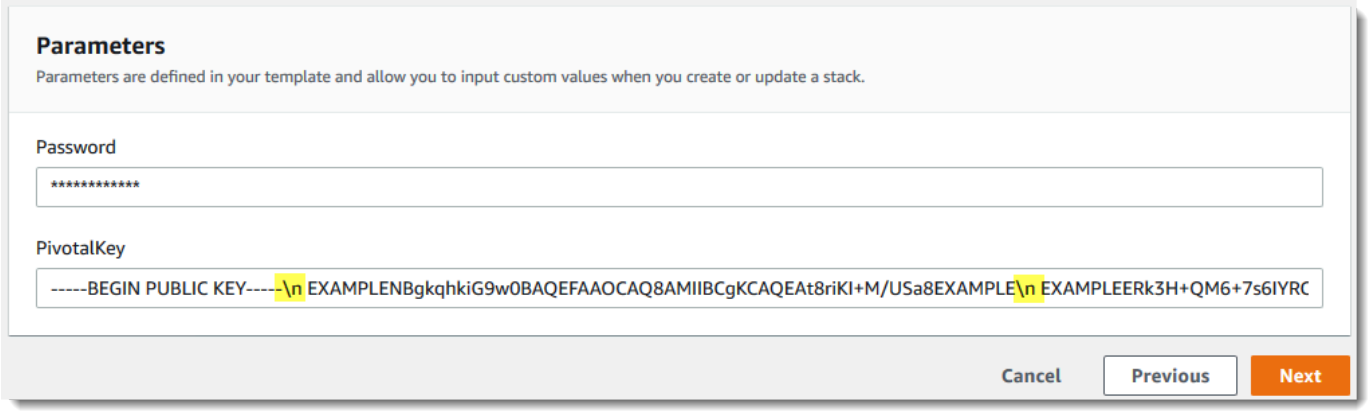
1. Accedi AWS Management Console e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com /cloudformazione.
2. Nella AWS CloudFormation home page, scegli Crea stack.
3. In Prerequisito: prepara il modello, se utilizzi il AWS CloudFormation modello di esempio, scegli Il modello è pronto.
4. In Specify template (Specifica modello), scegliere l'origine del modello. Per questa procedura dettagliata, scegli Carica un file modello e carica un AWS CloudFormation modello che crei un server Chef Automate. Cercare il file di modello e quindi scegliere Avanti.

   Un AWS CloudFormation modello può essere in formato YAML o JSON. È disponibile un AWS CloudFormation modello di esempio da utilizzare; assicurati di sostituire i valori di esempio con i tuoi. Puoi utilizzare lo strumento di progettazione del AWS CloudFormation modello per creare un nuovo modello o convalidarne uno esistente. Per ulteriori informazioni su come eseguire questa operazione, consulta la sezione relativa alla panoramica dell'interfaccia di progettazione AWS CloudFormation nella Guida per l'utente AWS CloudFormation .



5. Nella pagina Specify stack details (Specifica dettagli dello stack), immettere un nome per lo stack. Questo sarà diverso dal nome del server, è solo un nome di stack. Nell'area Parameters (Parametri), incollare i valori creati in the section called “Prerequisiti”. Inserire la password in Password.

   Incolla il contenuto del file della chiave RSA. PivotalKey Nella AWS CloudFormation console, è necessario aggiungere caratteri newline (**\n**) alla fine di ogni riga del valore della chiave pivotal, come mostrato nella schermata seguente. Scegli Next (Successivo).



6. Nella pagina Configura le opzioni dello stack, puoi aggiungere tag al server che stai creando con lo stack e scegliere un ruolo IAM per la creazione di risorse se non hai già specificato un ruolo IAM da utilizzare nel tuo modello. Dopo aver specificato le opzioni, scegliere Next (Avanti). Per ulteriori informazioni sulle opzioni avanzate come i rollback trigger, consulta Setting AWS CloudFormation Stack Options nella Guida per l'utente.AWS CloudFormation
7. Nella pagina Review (Esamina), esaminare le proprie opzioni. Quando si è pronti per creare lo stack del server, scegliere Create stack (Crea stack).

   Mentre aspetti di AWS CloudFormation creare lo stack, visualizza lo stato di creazione dello stack. Se si verifica un errore di creazione dello stack, esaminare i messaggi di errore visualizzati nella console per risolvere i problemi. Per ulteriori informazioni sulla risoluzione dei problemi relativi a errori dello stack AWS CloudFormation , consulta la sezione relativa alla risoluzione dei problemi relativi agli errori nella Guida per l'utente AWS CloudFormation .

   Al termine della creazione del server, il server Chef Automate sarà disponibile nella home page di AWS OpsWorks for Chef Automate , con lo stato online (in linea). Generare un nuovo Starter Kit e le credenziali del pannello di controllo di Chef Automate dalla pagina Properties (Proprietà) del server. Dopo che il server è online, il pannello di controllo Chef Automate è disponibile nel dominio del server, per una URL nel formato seguente: `https://`*`your_server_name`*`-`*`randomID`*`.`*`region`*`.opsworks-cm.io`.

   > **Note**
   >
   > Se hai specificato un dominio, un certificato e una chiave privata personalizzati per il tuo server, crea una voce CNAME nello strumento di gestione DNS dell'azienda che mappa il dominio personalizzato all'endpoint generato AWS OpsWorks for Chef Automate automaticamente per il server. Non è possibile gestire il server o connettersi al pannello di controllo Chef Automate del server finché non si esegue la mappatura dell'endpoint generato al valore del dominio personalizzato.
   > Per ottenere il valore dell'endpoint generato, esegui il AWS CLI comando seguente dopo che il server è online:
   >
   > ```
   > aws opsworks describe-servers --server-name server_name
   > ```

Creazione di un server Chef Automate tramite la CLI AWS CloudFormation

Se sul tuo computer locale non è già in esecuzione AWS CLI, scaricalo e installalo AWS CLI seguendo le istruzioni di installazione nella AWS Command Line Interface User Guide. Questa sezione non descrive tutti i parametri che è puoi utilizzare con il comando create-stack. Per ulteriori informazioni sui parametri create-stack, consulta create-stack nella documentazione di riferimento delle AWS CLI .

1. Assicurati di completare Prerequisiti per la creazione di un server AWS OpsWorks for Chef Automate .
2. Crea un ruolo di servizio e un profilo di istanza.

   Trova e copia i ruoli ARNs di servizio nel tuo account.

   ```
   aws iam list-roles --path-prefix "/service-role/" --no-paginate
   ```

   Nei risultati del comando `list-roles`, cercare le voci del ruolo di servizio e del profilo dell'istanza che sono simili a quanto segue. Prendi nota del ruolo ARNs di servizio e del profilo dell'istanza e aggiungili al AWS CloudFormation modello che stai utilizzando per creare lo stack di server.

   ```
   {
       "AssumeRolePolicyDocument": {
           "Version": "2012-10-17",
           "Statement": [
               {
                   "Action": "sts:AssumeRole",
                   "Effect": "Allow",
                   "Principal": {
                       "Service": "ec2.amazonaws.com"
                   }
               }
           ]
       },
       "RoleId": "AROZZZZZZZZZQG6R22HC",
       "CreateDate": "2018-01-05T20:42:20Z",
       "RoleName": "aws-opsworks-cm-ec2-role",
       "Path": "/service-role/",
       "Arn": "arn:aws:iam::000000000000:role/service-role/aws-opsworks-cm-ec2-role"
   },
   {
   ```

```
    "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "opsworks-cm.amazonaws.com"
                }
            }
        ]
    },
    "RoleId": "AROZZZZZZZZZZZZZZZ6QE",
    "CreateDate": "2018-01-05T20:42:20Z",
    "RoleName": "aws-opsworks-cm-service-role",
    "Path": "/service-role/",
    "Arn": "arn:aws:iam::000000000000:role/service-role/aws-opsworks-cm-service-
role"
}
```

3. Crea il AWS OpsWorks for Chef Automate server eseguendo nuovamente il create-stack comando.

   - Sostituiscilo *`stack_name`* con il nome del tuo stack. Questo è il nome dello AWS CloudFormation stack, non del server Chef Automate. Il nome del server Chef Automate è il valore di `ServerName` nel tuo modello. AWS CloudFormation
   - Sostituiscilo *`template`* con il percorso del file modello e l'estensione *`yaml or json`* con `.yaml` o `.json` come appropriato.
   - I valori di `--parameters` corrispondono a EngineAttributesdall'CreateServerAPI. Per Chef, gli attributi del motore forniti dall'utente per la creazione di un server sono `CHEF_AUTOMATE_PIVOTAL_KEY`, una chiave pubblica RSA codificata base64 generata mediante le utilità descritte in the section called “Prerequisiti” e `CHEF_AUTOMATE_ADMIN_PASSWORD`, una password compresa tra 8 e 32 caratteri creata dall'utente. Per ulteriori informazioni su `CHEF_AUTOMATE_ADMIN_PASSWORD`, consulta Crea un server Chef Automate utilizzando il AWS CLI. È possibile fornire un puntatore al file PEM che contiene la chiave pivotal come valore del parametro `PivotalKey`, come indicato nell'esempio. Se i valori per `CHEF_AUTOMATE_ADMIN_PASSWORD` e non `CHEF_AUTOMATE_PIVOTAL_KEY` sono specificati nel modello, è necessario fornire i valori nel AWS CLI comando.

```
aws cloudformation create-stack --stack-name stack_name
 --template-body file://template.yaml or json --parameters
 ParameterKey=PivotalKey,ParameterValue="base64_encoded_RSA_public_key_value"
```

Di seguito è riportato un esempio che include valori di esempio per gli attributi `CHEF_AUTOMATE_ADMIN_PASSWORD` e `CHEF_AUTOMATE_PIVOTAL_KEY`. Eseguite un comando simile se non avete specificato i valori per questi attributi nel AWS CloudFormation modello.

```
aws cloudformation create-stack --stack-name "OpsWorksCMChefServerStack"
 --template-body file://opsworkscm-server.yaml --parameters
 ParameterKey=PivotalKey,ParameterValue="$(openssl rsa -in "pivotalKey.pem" -
pubout)" ParameterKey=Password,ParameterValue="SuPer\$ecret890"
```

4. Al termine della creazione dello stack, apri la pagina delle proprietà del nuovo server nella AWS OpsWorks for Chef Automate console e scarica uno starter kit. Effettuare il download di un nuovo starter kit reimposta la password amministratore del pannello di controllo di Chef Automate.
5. Se il server utilizzerà un dominio, un certificato e una chiave privata personalizzati, attenersi alla procedura per la configurazione `knife.rb` in (Opzionale) Configurare `knife` per l'utilizzo di un dominio personalizzato e quindi passare alla fase 7.
6. Per utilizzare i comandi `knife` nel nuovo server, aggiornare le impostazioni del file di configurazione di Chef `knife.rb`. Con lo starter kit è incluso un file `knife.rb` di esempio. Nell'esempio seguente viene illustrato come impostare `knife.rb` su un server che non utilizza un dominio personalizzato. Se si utilizza un dominio personalizzato, consultare (Opzionale) Configurare `knife` per l'utilizzo di un dominio personalizzato per le istruzioni di configurazione `knife`.

   - Sostituisci *`ENDPOINT`* con il valore dell'endpoint del server. Questo è parte dell'output dell'operazione di creazione dello stack. Puoi ottenere l'endpoint eseguendo il seguente comando.

     ```
     aws cloudformation describe-stacks --stack-name stack_name
     ```

   - Sostituiscilo *`key_pair_file.pem`* nella `client_key` configurazione con il nome del file PEM che contiene il file `CHEF_AUTOMATE_PIVOTAL_KEY` che hai usato per creare il tuo server.

```
base_dir = File.join(File.dirname(File.expand_path(__FILE__)), '..')

log_level                :info
log_location             STDOUT
node_name                'pivotal'
client_key               File.join(base_dir, '.chef', 'key_pair_file.pem')
syntax_check_cache_path  File.join(base_dir, '.chef', 'syntax_check_cache')
cookbook_path            [File.join(base_dir, 'cookbooks')]

chef_server_url          'ENDPOINT/organizations/default'
ssl_ca_file              File.join(base_dir, '.chef', 'ca_certs', 'opsworks-cm-
ca-2020-root.pem')
trusted_certs_dir        File.join(base_dir, '.chef', 'ca_certs')
```

7. Al termine del processo di creazione del server, vai su the section called “Terminare la configurazione e caricare i libri di ricette”. Se si verifica un errore di creazione dello stack, esaminare i messaggi di errore visualizzati nella console per risolvere i problemi. Per ulteriori informazioni sulla risoluzione degli errori negli AWS CloudFormation stack, consulta Risoluzione dei problemi degli errori nella Guida per l'AWS CloudFormation utente.

Aggiornare un AWS OpsWorks for Chef Automate server per utilizzare un dominio personalizzato

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Questa sezione descrive come aggiornare un AWS OpsWorks for Chef Automate server esistente per utilizzare un dominio e un certificato personalizzati utilizzando un backup del server per creare un nuovo server. In sostanza, si sta copiando un server AWS OpsWorks for Chef Automate 2.0 esistente creando un nuovo server da un backup, quindi configurando il nuovo server per utilizzare un dominio, un certificato e una chiave privata personalizzati.

Argomenti

- Prerequisiti
- Limitazioni
- Aggiornare un server per utilizzare un dominio personalizzato
- Vedi anche

Prerequisiti

Di seguito sono riportati i requisiti per l'aggiornamento di un AWS OpsWorks for Chef Automate server esistente per utilizzare un dominio e un certificato personalizzati.

- Il server che si desidera aggiornare (o copiare) deve eseguire Chef Automate 2.0.
- Decidere quale backup si desidera utilizzare per creare un nuovo server. È necessario avere almeno un backup disponibile del server che si desidera aggiornare. Per ulteriori informazioni sui backup in AWS OpsWorks for Chef Automate, vedereEseguire il backup di un server AWS OpsWorks for Chef Automate.
- Prepara il ruolo di servizio e il profilo di istanza ARNs che hai usato per creare il server esistente che è l'origine del backup.
- Assicurarsi di utilizzare la versione più recente della AWS CLI. Per ulteriori informazioni sull'aggiornamento AWS CLI degli strumenti, consulta la Guida per l'utente dell'interfaccia AWS CLI a riga di comando di AWS (Installazione della Guida per l'utente dell'interfaccia a riga di comando di AWS).

Limitazioni

Quando aggiorni un server esistente creando un nuovo server da un backup, il nuovo server non può essere esattamente lo stesso del AWS OpsWorks for Chef Automate server esistente.

- È possibile completare questa procedura solo utilizzando AWS CLI o uno dei AWS SDKs. Non è possibile creare un nuovo server da un backup utilizzando la AWS Management Console.
- Il nuovo server non può utilizzare lo stesso nome del server esistente all'interno di un account e all'interno di una regione AWS. Il nome deve essere diverso da quello del server esistente utilizzato come origine del backup.
- I nodi collegati al server esistente non vengono gestiti dal nuovo server. È necessario effettuare una delle seguenti operazioni.
  - Collegare nodi diversi, perché i nodi non possono essere gestiti da più server Chef Automate.

- Eseguire la migrazione dei nodi dal server esistente (l'origine del backup) al nuovo server e al nuovo endpoint di dominio personalizzato. Per ulteriori informazioni sulla migrazione dei nodi, vedere la documentazione di Chef.

Aggiornare un server per utilizzare un dominio personalizzato

Per aggiornare un server Chef Automate 2.0 esistente, è possibile crearne una copia eseguendo il comando `create-server`, aggiungendo parametri per specificare un backup, un dominio personalizzato, un certificato personalizzato e una chiave privata personalizzata.

1. Se non disponi di un ruolo di servizio o di un profilo ARNs di istanza da specificare nel `create-server` comando, segui i passaggi da 1 Crea un server Chef Automate utilizzando il AWS CLI a 5 per creare un ruolo di servizio e un profilo di istanza da utilizzare.
2. Se non è stato già fatto, trovare il backup del server Chef Automate 2.0 esistente su cui si desidera basare un nuovo server con un dominio personalizzato. Esegui il comando seguente per mostrare le informazioni su tutti i AWS OpsWorks for Chef Automate backup nel tuo account e in una regione. Assicurarsi di annotare l'ID del backup che si desidera utilizzare.

   ```
   aws opsworks-cm --region region name describe-backups
   ```

3. Crea il AWS OpsWorks for Chef Automate server eseguendo il `create-server` comando.

   - Il valore `--engine` è `ChefAutomate`, `--engine-model` è `Single` e `--engine-version` è `12`.
   - Il nome del server deve essere univoco all'interno del tuo AWS account, all'interno di ogni regione. I nomi dei server devono iniziare con una lettera; quindi sono consentiti lettere, numeri o trattini (-) fino a un massimo di 40 caratteri.
   - Utilizzare il profilo di istanza ARN e il ruolo di servizio ARN dalla fase 1.
   - I tipi di istanza validi sono `m5.large`, `r5.xlarge` o `r5.2xlarge`. Per ulteriori informazioni sulle specifiche di questi tipi di istanze, consulta Instance Types nella Amazon EC2 User Guide.
   - Il parametro `--engine-attributes` è opzionale; se non specifichi uno o entrambi i valori, il processo di creazione del server genera i valori al posto tuo. Se aggiungi `--engine-attributes`, specifica il valore `CHEF_AUTOMATE_PIVOTAL_KEY` che hai generato nella Fase 2, un valore di `CHEF_AUTOMATE_ADMIN_PASSWORD` o entrambi.

Se non imposti un valore per `CHEF_AUTOMATE_ADMIN_PASSWORD`, viene generata e restituita una password come parte della risposta `create-server`. Inoltre puoi scaricare nuovamente lo starter kit nella console, il quale rigenera questa password. La lunghezza della password è compresa tra 8 e 32 caratteri. La password può contenere lettere, numeri e caratteri speciali (`!/@#$%^+=_`). La password deve contenere almeno una lettera minuscola, una maiuscola, un numero e un carattere speciale.

- Una coppia di chiavi SSH è facoltativa, ma può essere utile per connettersi al server Chef Automate se dovessi reimpostare la password amministratore del pannello di controllo di Chef Automate. Per ulteriori informazioni sulla creazione di una coppia di chiavi SSH, consulta Amazon EC2 Key Pairs nella Amazon EC2 User Guide.
- Per utilizzare un dominio personalizzato, aggiungere i seguenti parametri al comando. Diversamente, il processo di creazione del server Chef Automate genera automaticamente un endpoint. Tutti e tre i parametri sono necessari per configurare un dominio personalizzato. Per informazioni sui requisiti aggiuntivi per l'utilizzo di questi parametri, consulta CreateServeril AWS OpsWorks CM API Reference.
  - `--custom-domain` - Un endpoint pubblico opzionale di un server, ad esempio `https://aws.my-company.com`.
  - `--custom-certificate` - Un certificato HTTPS in formato PEM. Il valore può essere un singolo certificato autofirmato o una catena di certificati.
  - `--custom-private-key` - Una chiave privata in formato PEM per la connessione al server tramite HTTPS. La chiave privata non deve essere crittografata; non può essere protetta da password o da passphrase.
- La manutenzione del sistema settimanale è obbligatoria. I valori validi devono essere specificati nel seguente formato: `DDD:HH:MM`. L'ora specificata è espressa nel Tempo coordinato universale (UTC). Se non si specifica un valore per `--preferred-maintenance-window`, il valore predefinito è un periodo casuale di un'ora il martedì, mercoledì o venerdì.
- I valori validi per `--preferred-backup-window` devono essere specificati in uno dei seguenti formati: `HH:MM` per backup giornaliero o `DDD:HH:MM` per backup settimanali. L'orario specificato è nel formato UTC. Il valore predefinito è un'ora di avvio giornaliera casuale. Per disattivare i backup automatici, aggiungere invece il parametro `--disable-automated-backup`.
- Per`--security-group-ids`, inserisci uno o più gruppi di sicurezza IDs, separati da uno spazio.

- Per `--subnet-ids`, inserire un ID di sottorete.
- Per `--backup-id`, immettere l'ID del backup copiato nella fase 2.

```
aws opsworks-cm create-server --engine "ChefAutomate" --engine-model "Single"
 --engine-version "12" --server-name "server_name" --instance-profile-arn
 "instance_profile_ARN" --instance-type "instance_type" --engine-attributes
 '{"CHEF_AUTOMATE_PIVOTAL_KEY":"pivotal_key","CHEF_AUTOMATE_ADMIN_PASSWORD":"password"}'
 --key-pair "key_pair_name" --preferred-maintenance-window
 "ddd:hh:mm" --preferred-backup-window "ddd:hh:mm" --security-group-
ids security_group_id1 security_group_id2 --service-role-arn "service_role_ARN" --
subnet-ids subnet_ID --backup-id backup_ID
```

Nell'esempio seguente viene creato un server Chef Automate che utilizza un dominio personalizzato.

```
aws opsworks-cm create-server --engine "ChefAutomate" --engine-model "Single" --
engine-version "12" \
    --server-name "my-custom-domain-server" \
    --instance-profile-arn "arn:aws:iam::12345678912:instance-profile/aws-opsworks-
cm-ec2-role" \
    --instance-type "m5.large" \
    --engine-attributes
 '{"CHEF_AUTOMATE_PIVOTAL_KEY":"MZZE...Wobg","CHEF_AUTOMATE_ADMIN_PASSWORD":"zZZzDj2DLYXSZF
 \
    --custom-domain "my-chef-automate-server.my-corp.com" \
    --custom-certificate "-----BEGIN CERTIFICATE----- EXAMPLEqEXAMPLE== -----END
 CERTIFICATE-----" \
    --custom-private-key "-----BEGIN RSA PRIVATE KEY----- EXAMPLEqEXAMPLE= -----END
 RSA PRIVATE KEY-----" \
    --key-pair "amazon-test" \
    --preferred-maintenance-window "Mon:08:00" \
    --preferred-backup-window "Sun:02:00" \
    --security-group-ids sg-b00000001 sg-b0000008 \
    --service-role-arn "arn:aws:iam::12345678912:role/service-role/aws-opsworks-cm-
service-role" \
    --subnet-ids subnet-300aaa00 \
    --backup-id MyChefServer-20191004122143125
```

4. AWS OpsWorks for Chef Automate impiega circa 15 minuti per creare un nuovo server. Nell'output del comando `create-server`, copiare il valore dell'attributo `Endpoint`. Di seguito è riportato un esempio.

   ```
   "Endpoint": "automate-07-exampleexample.opsworks-cm.us-east-1.amazonaws.com"
   ```

   Non annullare l'output del comando `create-server` né chiudere la sessione shell, poiché l'output può contenere informazioni importanti che non vengono visualizzate nuovamente. Per ottenere le password e lo starter kit dai risultati `create-server`, passare alla fase successiva.

5. Se hai scelto di AWS OpsWorks for Chef Automate generare automaticamente una chiave e una password, puoi estrarle in formati utilizzabili dai `create-server` risultati utilizzando un processore JSON come jq. Dopo aver installato jq, puoi eseguire i seguenti comandi per estrarre la chiave pivotal, la password amministratore del pannello di controllo di Chef Automate e lo starter kit. Se non hai fornito la chiave pivotal e la password nella Fase 3, assicurati di salvare in una posizione comoda e sicura la chiave pivotal e la password amministratore estratte.

   ```
   #Get the Chef password:
   cat resp.json | jq -r '.Server.EngineAttributes[] | select(.Name ==
    "CHEF_AUTOMATE_ADMIN_PASSWORD") | .Value'

   #Get the Chef Pivotal Key:
   cat resp.json | jq -r '.Server.EngineAttributes[] | select(.Name ==
    "CHEF_AUTOMATE_PIVOTAL_KEY") | .Value'

   #Get the Chef Starter Kit:
   cat resp.json | jq -r '.Server.EngineAttributes[] | select(.Name ==
    "CHEF_STARTER_KIT") | .Value' | base64 -D > starterkit.zip
   ```

6. Facoltativamente, se non avete estratto lo starter kit dai risultati dei `create-server` comandi, potete scaricare un nuovo starter kit dalla pagina Proprietà del server nella console. AWS OpsWorks for Chef Automate Effettuare il download di un nuovo starter kit reimposta la password amministratore del pannello di controllo di Chef Automate.

7. Create una voce CNAME nello strumento di gestione DNS aziendale per indirizzare il dominio personalizzato all' AWS OpsWorks for Chef Automate endpoint copiato nel passaggio 4. Non è possibile raggiungere o eseguire l'accesso al server fino a quando non si completa questa fase.

8. Al termine del processo di creazione del server, vai su the section called “Terminare la configurazione e caricare i libri di ricette”.

Vedi anche

- Crea un server Chef Automate utilizzando il AWS CLI
- Ripristino di un AWS OpsWorks for Chef Automate server da un backup
- CreateServernel CM API Reference AWS OpsWorks
- create-server in Riferimento ai comandi AWS CLI

Rigenera lo starter kit per un server AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Lo starter kit per AWS OpsWorks for Chef Automate contiene un file README con esempi, un file di `knife.rb` configurazione e una chiave privata per l'utente primario o principale. Ogni volta che scarichi lo starter kit viene generata una nuova coppia di chiavi e la vecchia chiave viene ripristinata. È possibile rigenerare lo starter kit per un server in due modi: AWS OpsWorks for Chef Automate

- Nella AWS OpsWorks console, nel menu Azioni della pagina dei dettagli di un AWS OpsWorks for Chef Automate server. Ti viene richiesto di confermare se desideri rigenerare e reimpostare la vecchia chiave pivotal.
- Eseguendo i comandi in. AWS CLI

Per ulteriori informazioni su come utilizzare lo starter kit, vedereConfigura il server Chef utilizzando lo Starter Kit.

Rigenera lo AWS OpsWorks for Chef Automate starter kit con AWS CLI

> **Note**
>
> Quando rigeneri lo starter kit, rigeneri e ripristini anche la coppia di chiavi di autenticazione per il tuo server Chef Automate ed elimini la coppia di chiavi corrente.

Rigenera lo starter kit eseguendo il comando. update-server-engine-attributes In una AWS CLI sessione, esegui il comando seguente. Specificate il nome del server come valore di`--server-name`. Per impostare una chiave pubblica personale come valore di`CHEF_AUTOMATE_PIVOTAL_KEY`, specifica il valore della chiave pubblica in`--attribute-value`. Altrimenti, imposta `--attribute-value` su null.

```
aws opsworks-cm update-server-engine-attributes \
   --server-name server_name \
   --attribute-name "CHEF_AUTOMATE_PIVOTAL_KEY" \
   --attribute-value your_public_key
```

Il comando seguente è un esempio che specifica un valore di chiave pubblica che l'amministratore del server desidera utilizzare.

```
aws opsworks-cm update-server-engine-attributes \
   --server-name your-test-server \
   --attribute-name "CHEF_AUTOMATE_PIVOTAL_KEY" \
   --attribute-value "-----BEGIN PUBLIC KEY-----ExamplePublicKey-----END PUBLIC
 KEY-----"
```

Il comando seguente è un esempio che consente di AWS OpsWorks for Chef Automate rigenerare la chiave pubblica.

```
aws opsworks-cm update-server-engine-attributes \
   --server-name your-test-server \
   --attribute-name "CHEF_AUTOMATE_PIVOTAL_KEY" \
   --attribute-value null
```

L'output di questo comando è costituito da informazioni sul server e da un file ZIP con codifica Base64. Il file ZIP contiene uno starter kit Chef, che include un README, un file di configurazione e

la chiave privata RSA richiesta. Salva questo file, decomprimilo e poi passa alla directory in cui hai decompresso il contenuto del file. Da questa directory, puoi eseguire comandi. `knife`

Lavorare con i tag sulle AWS OpsWorks for Chef Automate risorse

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

I tag sono parole o frasi che fungono da metadati per l'identificazione e l'organizzazione delle risorse AWS. In AWS OpsWorks for Chef Automate, una risorsa può avere fino a 50 tag applicati dall'utente. Ciascun tag è formato da una chiave e da un valore opzionale. È possibile applicare tag alle seguenti risorse in AWS OpsWorks for Chef Automate:

- AWS OpsWorks for Chef Automate server
- Backup dei server AWS OpsWorks for Chef Automate

I tag sulle AWS risorse possono aiutarti a tenere traccia dei costi, controllare l'accesso alle risorse, raggruppare le risorse per automatizzare le attività o organizzare le risorse per scopo o fase del ciclo di vita. Per ulteriori informazioni sui vantaggi dei tag, consulta Strategie di tagging in AWS Answers e Utilizzo di tag per l'allocazione costi nella Guida utente di AWS Billing and Cost Management .

Per utilizzare i tag per controllare l'accesso ai AWS OpsWorks for Chef Automate server o ai backup, è possibile creare o modificare le dichiarazioni di policy in AWS Identity and Access Management (IAM). Per ulteriori informazioni, consulta Controllo dell'accesso alle risorse AWS mediante i tag delle risorse nella Guida utente di AWS Identity and Access Management .

Quando si applicano tag a un AWS OpsWorks for Chef Automate server, i tag vengono applicati anche ai backup del server, al bucket Amazon S3 che archivia i backup, all'istanza EC2 Amazon del server, ai segreti per il server AWS Secrets Manager in cui sono archiviati e all'indirizzo IP elastico utilizzato dal server. I tag non vengono propagati AWS CloudFormation nello stack utilizzato per creare il server. AWS OpsWorks

Argomenti

- Come funzionano i tag AWS OpsWorks for Chef Automate
- Aggiungi e gestisci tag in AWS OpsWorks for Chef Automate (Console)
- Aggiungi e gestisci tag in AWS OpsWorks for Chef Automate (CLI)
- Vedi anche

Come funzionano i tag AWS OpsWorks for Chef Automate

In questa versione, è possibile aggiungere e gestire i tag utilizzando l'API CM AWS OpsWorks o AWS Management Console. AWS OpsWorks CM tenta inoltre di aggiungere tag aggiunti a un server alle AWS risorse associate al server, tra cui l' EC2 istanza, i segreti in Secrets Manager, l'indirizzo IP elastico, il gruppo di sicurezza, il bucket S3 e i backup. Nella tabella seguente viene fornita una panoramica delle modalità di aggiunta e gestione dei tag in AWS OpsWorks for Chef Automate.

Azione	Cosa usare
Aggiungi tag a un nuovo AWS OpsWorks for Chef Automate server o a un backup che stai creando manualmente.	• Scegliere Crea server Chef Automate e aggiungere tag nella pagina Configura impostazioni avanzate. • Scegliere Crea backup nella pagina Backup per un server esistente e aggiungere tag nella pagina Crea un backup del server Chef Automate 2 . • Aggiungere un parametro `Tags` ai comandi `CreateServer` o `CreateBackup` .
Visualizza i tag su una risorsa.	• Nella pagina dei dettagli del server scegliere Tag nel riquadro di spostamento. • Nella pagina Backup del server selezionare un backup, quindi scegliere Modifica backup. • Esegui il comando `ListTagsForResource` .
Aggiungi tag a un AWS OpsWorks for Chef Automate server esistente o a un backup, indipendentemente dal fatto che il backup sia stato creato manualmente o automaticamente.	• Nella pagina dei dettagli del server scegliere Tag nel riquadro di spostamento, quindi scegliere Modifica.

Azione	Cosa usare
	• Nella pagina Backup del server selezionare un backup, quindi scegliere Modifica backup. • Esegui il comando `TagResource`.
Per eliminare tag da una risorsa	• Nella pagina dei dettagli del server scegliere Tag nel riquadro di spostamento, quindi scegliere Modifica. Scegliere X accanto ai tag che si desidera eliminare. • Nella pagina Backup del server selezionare un backup, quindi scegliere Modifica backup. Scegliere X accanto ai tag che si desidera eliminare. • Esegui il comando `UntagResource`.

Le risposte `DescribeServers` e `DescribeBackups` non includono informazioni sui tag. Per visualizzare i tag, utilizzare l'API `ListTagsForResource`.

Aggiungi e gestisci tag in AWS OpsWorks for Chef Automate (Console)

Le procedure in questa sezione vengono eseguite in AWS Management Console.

Se si aggiungono tag, una chiave tag non può essere vuota. La chiave può avere un massimo di 127 caratteri e può contenere solo lettere, numeri o separatori Unicode oppure i seguenti caratteri speciali: `+ - = . _ : / @` i valori dei tag sono opzionali. È possibile aggiungere un tag che ha una chiave, ma senza valori. Il valore può avere un massimo di 255 caratteri e può contenere solo lettere, numeri o separatori Unicode oppure i seguenti caratteri speciali: `+ - = . _ : / @`.

Argomenti

- Aggiungi tag a un nuovo AWS OpsWorks for Chef Automate server (console)
- Aggiungere tag a un nuovo backup (console)
- Aggiungere o visualizzare tag su un server esistente (Console)
- Aggiungere o visualizzare tag su un backup esistente (Console)
- Eliminare tag da un server (Console)
- Eliminare tag da un backup (Console)

Aggiungi tag a un nuovo AWS OpsWorks for Chef Automate server (console)

1. Assicurati di completare tutti i prerequisiti per la creazione di un AWS OpsWorks for Chef Automate server.
2. Seguire i passaggi 1-10 in Creare un server Chef Automate.
3. Dopo aver specificato le impostazioni di backup automatizzato, aggiungere i tag nell'area Tag della pagina Configura impostazioni avanzate . È possibile aggiungere un massimo di 50 tag. Una volta aggiunti i tag, scegliere Next (Avanti).
4. Passare al passaggio 13 di Creare un server Chef Automate, ed esaminare le impostazioni scelte per il nuovo server.

Aggiungere tag a un nuovo backup (console)

1. Nella AWS OpsWorks for Chef Automate home page, scegli un server Chef Automate esistente.
2. Nella pagina dei dettagli del server scegliere Backup nel riquadro di spostamento.
3. Nella pagina Backup scegliere Crea backup.
4. Aggiungi tag. Al termine dell'aggiunta delle regole, scegliere Crea.

Aggiungere o visualizzare tag su un server esistente (Console)

1. Nella AWS OpsWorks for Chef Automate home page, scegli un server Chef Automate esistente per aprirne la pagina dei dettagli.
2. Scegliere Tag nel riquadro di spostamento oppure, nella parte inferiore della pagina dei dettagli, scegliere Visualizza tutti i tag.
3. Nella pagina Tag scegliere Modifica.
4. Aggiungere o modificare tag sul server. Al termine, selezionare Save (Salva).

> **Note**
>
> Tieni presente che la modifica dei tag sul tuo server Chef Automate modifica anche i tag sulle risorse associate al server, come l' EC2 istanza, l'indirizzo IP elastico, il gruppo di sicurezza, il bucket S3 e i backup.

Aggiungere o visualizzare tag su un backup esistente (Console)

1. Nella AWS OpsWorks for Chef Automate home page, scegli un server Chef Automate esistente per aprirne la pagina dei dettagli.
2. Scegliere Backup nel riquadro di spostamento oppure, nell'area Backup recenti della pagina dei dettagli, scegliere Visualizza tutti i backup.
3. Nella pagina Backup scegliere un backup da gestire, quindi scegliere Modifica backup.
4. Aggiungere o modificare tag sul backup. Scegliere Aggiorna al termine.

Eliminare tag da un server (Console)

1. Nella AWS OpsWorks for Chef Automate home page, scegli un server Chef Automate esistente per aprirne la pagina dei dettagli.
2. Scegliere Tag nel riquadro di spostamento oppure, nella parte inferiore della pagina dei dettagli, scegliere Visualizza tutti i tag.
3. Nella pagina Tag scegliere Modifica.
4. Scegliere X accanto a un tag per eliminare il tag. Al termine, selezionare Save (Salva).

> **Note**
>
> Tieni presente che la modifica dei tag sul tuo server Chef Automate modifica anche i tag sulle risorse associate al server, come l' EC2 istanza, l'indirizzo IP elastico, il gruppo di sicurezza, il bucket S3 e i backup.

Eliminare tag da un backup (Console)

1. Nella AWS OpsWorks for Chef Automate home page, scegli un server Chef Automate esistente per aprirne la pagina dei dettagli.
2. Scegliere Backup nel riquadro di spostamento oppure, nell'area Backup recenti della pagina dei dettagli, scegliere Visualizza tutti i backup.
3. Nella pagina Backup scegliere un backup da gestire, quindi scegliere Modifica backup.
4. Scegliere X accanto a un tag per eliminare il tag. Scegliere Aggiorna al termine.

Aggiungi e gestisci tag in AWS OpsWorks for Chef Automate (CLI)

Le procedure in questa sezione vengono eseguite in AWS CLI. Assicurati di utilizzare l'ultima versione di AWS CLI prima di iniziare a lavorare con i tag. Per ulteriori informazioni sull'installazione o l'aggiornamento di AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Se si aggiungono tag, una chiave tag non può essere vuota. La chiave può avere un massimo di 127 caratteri e può contenere solo lettere, numeri o separatori Unicode oppure i seguenti caratteri speciali: `+ - = . _ : / @` i valori dei tag sono opzionali. È possibile aggiungere un tag che ha una chiave, ma senza valori. Il valore può avere un massimo di 255 caratteri e può contenere solo lettere, numeri o separatori Unicode oppure i seguenti caratteri speciali: `+ - = . _ : / @`.

Argomenti

- Aggiungere tag a un nuovo AWS OpsWorks for Chef Automate server (CLI)
- Aggiungere tag a un nuovo backup (CLI)
- Aggiungere tag a server o backup esistenti (CLI)
- Elenca i tag delle risorse
- Per eliminare tag da una risorsa

Aggiungere tag a un nuovo AWS OpsWorks for Chef Automate server (CLI)

È possibile utilizzare il AWS CLI per aggiungere tag quando si crea un AWS OpsWorks for Chef Automate server. Questa procedura non descrive in modo completo come creare un server. Per informazioni dettagliate su come creare un AWS OpsWorks for Chef Automate server utilizzando, AWS CLI Crea un server Chef Automate utilizzando il AWS CLI consulta questa guida. È possibile aggiungere fino a 50 tag a un server.

1. Assicurati di completare tutti i prerequisiti per la creazione di un AWS OpsWorks for Chef Automate server.
2. Completa i passaggi 1-5 di Crea un server Chef Automate utilizzando il AWS CLI.
3. Per il passaggio 6, quando si esegue il comando `create-server`, aggiungere il parametro `--tags` al comando, come illustrato nell'esempio seguente.

```
aws opsworks-cm create-server ... --tags Key=Key1,Value=Value1
 Key=Key2,Value=Value2
```

Di seguito è riportato un esempio che mostra solo la parte dei tag del comando `create-server`.

```
aws opsworks-cm create-server ... --tags Key=Stage,Value=Production
 Key=Department,Value=Marketing
```

4. Completare le fasi restanti in Crea un server Chef Automate utilizzando il AWS CLI. Per verificare che i tag siano stati aggiunti al nuovo server, attenersi alla procedura descritta in Elenca i tag delle risorse in questo argomento.

Aggiungere tag a un nuovo backup (CLI)

È possibile AWS CLI utilizzarlo per aggiungere tag quando si crea un nuovo backup manuale di un AWS OpsWorks for Chef Automate server. Questa procedura non descrive in modo completo come creare un backup manuale. Per informazioni dettagliate su come creare un backup manuale, vedere «Per eseguire un backup manuale nel AWS CLI" inEseguire il backup di un server AWS OpsWorks for Chef Automate. È possibile aggiungere fino a 50 tag a un backup. Se un server dispone di tag, ai nuovi backup vengono automaticamente applicati i tag del server.

Per impostazione predefinita, quando si crea un nuovo AWS OpsWorks for Chef Automate server, i backup automatici sono abilitati. È possibile aggiungere tag a un backup automatico eseguendo il comando `tag-resource` descritto in Aggiungere tag a server o backup esistenti (CLI) in questo argomento.

- Per aggiungere tag a un backup manuale durante la creazione del backup, eseguire il comando seguente. Viene visualizzata solo la parte dei tag del comando. Per un esempio del comando `create-backup` completo, vedere «Per eseguire un backup manuale in AWS CLI" in Eseguire il backup di un server AWS OpsWorks for Chef Automate.

  ```
  aws opsworks-cm create-backup ... --tags Key=Key1,Value=Value1
   Key=Key2,Value=Value2
  ```

  L'esempio seguente mostra solo la parte tag del comando `create-backup`.

  ```
  aws opsworks-cm create-backup ... --tags Key=Stage,Value=Production
   Key=Department,Value=Marketing
  ```

Aggiungere tag a server o backup esistenti (CLI)

È possibile eseguire il comando `tag-resource` per aggiungere tag a server AWS OpsWorks for Chef Automate o backup esistenti (indipendentemente dal fatto che i backup siano stati creati automaticamente o manualmente). Specificare il numero di risorsa Amazon (ARN) di una risorsa di destinazione per aggiungervi dei tag.

1. Per ottenere l'ARN della risorsa a cui si desidera applicare i tag:

   - Per un server, eseguire `describe-servers --server-name` *`server_name`*. I risultati del comando mostrano il server ARN.
   - Per un backup, eseguire `describe-backups --backup-id` *`backup_ID`*. I risultati del comando mostrano l'ARN di backup. È inoltre possibile eseguire `describe-backups --server-name` *`server_name`* l'operazione per mostrare informazioni su tutti i backup per un server specifico AWS OpsWorks for Chef Automate .

   L'esempio seguente mostra solo `ServerArn` nei risultati di un comando `describe-servers --server-name opsworks-cm-test`. Il valore `ServerArn` viene aggiunto a un comando `tag-resource` per aggiungere tag al server.

   ```
   {
       "Servers": [
           {
               ...
               "ServerArn": "arn:aws:opsworks-cm:us-west-2:123456789012:server/
   opsworks-cm-test/EXAMPLEd-66b0-4196-8274-d1a2bEXAMPLE"
           }
       ]
   }
   ```

2. Eseguire il comando `tag-resource` con l'ARN restituito nel passaggio 1.

   ```
   aws opsworks-cm tag-resource --resource-arn "server_or_backup_ARN" --tags
    Key=Key1,Value=Value1 Key=Key2,Value=Value2
   ```

   Di seguito è riportato un esempio.

```
aws opsworks-cm tag-resource --resource-arn "arn:aws:opsworks-cm:us-
west-2:123456789012:server/opsworks-cm-test/EXAMPLEd-66b0-4196-8274-d1a2bEXAMPLE"
 --tags Key=Stage,Value=Production Key=Department,Value=Marketing
```

3. Per verificare che i tag siano stati aggiunti correttamente, passare alla procedura successiva Elenca i tag delle risorse.

Elenca i tag delle risorse

È possibile eseguire il `list-tags-for-resource` comando per mostrare i tag collegati ai AWS OpsWorks for Chef Automate server o ai backup. Specificare l'ARN di una risorsa di destinazione per visualizzarne i tag.

1. Per ottenere l'ARN della risorsa per la quale si desidera elencare i tag:
   - Per un server, eseguire `describe-servers --server-name` *`server_name`*. I risultati del comando mostrano il server ARN.
   - Per un backup, eseguire `describe-backups --backup-id` *`backup_ID`*. I risultati del comando mostrano l'ARN di backup. Puoi anche eseguire `describe-backups --server-name` *`server_name`* per mostrare informazioni su tutti i backup per un server specifico AWS OpsWorks for Chef Automate .
2. Eseguire il comando `list-tags-for-resource` con l'ARN restituito nel passaggio 1.

   ```
   aws opsworks-cm list-tags-for-resource --resource-arn "server_or_backup_ARN"
   ```

   Di seguito è riportato un esempio.

   ```
   aws opsworks-cm tag-resource --resource-arn "arn:aws:opsworks-cm:us-
   west-2:123456789012:server/opsworks-cm-test/EXAMPLEd-66b0-4196-8274-d1a2bEXAMPLE"
   ```

   Se sono presenti tag sulla risorsa, il comando restituisce risultati come i seguenti.

   ```
   {
       "Tags": [
           {
               "Key": "Stage",
               "Value": "Production"
           },
   ```

```
        {
            "Key": "Department",
            "Value": "Marketing"
        }
    ]
}
```

Per eliminare tag da una risorsa

È possibile eseguire il comando `untag-resource` per eliminare i tag da server AWS OpsWorks for Chef Automate o backup. Se la risorsa viene eliminata, anche i tag sulla risorsa vengono eliminati. Specificare l'Amazon Resource Number (ARN) di una risorsa di destinazione per rimuovere i tag da essa.

1. Per ottenere l'ARN della risorsa da cui si desidera rimuovere i tag:
   - Per un server, eseguire `describe-servers --server-name` *`server_name`*. I risultati del comando mostrano il server ARN.
   - Per un backup, eseguire `describe-backups --backup-id` *`backup_ID`*. I risultati del comando mostrano l'ARN di backup. Puoi anche eseguire l'operazione `describe-backups --server-name` *`server_name`* per mostrare informazioni su tutti i backup per un server specifico AWS OpsWorks for Chef Automate .
2. Eseguire il comando `untag-resource` con l'ARN restituito nel passaggio 1. Specificare solo i tag che si desidera eliminare.

   ```
   aws opsworks-cm untag-resource --resource-arn "server_or_backup_ARN" --tags
    Key=Key1,Value=Value1 Key=Key2,Value=Value2
   ```

   In questo esempio, il comando `untag-resource` rimuove solo il tag con una chiave `Stage` e un valore pari a `Production`.

   ```
   aws opsworks-cm untag-resource --resource-arn "arn:aws:opsworks-cm:us-
   west-2:123456789012:server/opsworks-cm-test/EXAMPLEd-66b0-4196-8274-d1a2bEXAMPLE"
    --tags Key=Stage,Value=Production
   ```

3. Per verificare che i tag siano stati eliminati correttamente, attenersi alla procedura descritta in Elenca i tag delle risorse in questo argomento.

Vedi anche

- Crea un server Chef Automate utilizzando il AWS CLI
- Eseguire il backup di un server AWS OpsWorks for Chef Automate
- AWS Tagging Strategies (Strategie di tagging AWS)
- Controllo dell'accesso alle AWS risorse utilizzando i tag delle risorse nella Guida per l'AWS Identity and Access Management utente
- Uso dei tag per l'allocazione dei costi nella Guida per l'utente di AWS Billing and Cost Management .
- CreateBackup in Documentazione di riferimento dell'API CM di AWS OpsWorks
- CreateServer in Documentazione di riferimento dell'API CM di AWS OpsWorks
- TagResource in Documentazione di riferimento dell'API CM di AWS OpsWorks
- ListTagsForResource in Documentazione di riferimento dell'API CM di AWS OpsWorks
- UntagResource in Documentazione di riferimento dell'API CM di AWS OpsWorks

Backup e ripristino di un AWS OpsWorks for Chef Automate server

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Questa sezione descrive come eseguire il backup e il ripristino di un AWS OpsWorks for Chef Automate server e come eliminare backup.

Argomenti

- Eseguire il backup di un server AWS OpsWorks for Chef Automate
- Ripristino di un AWS OpsWorks for Chef Automate server da un backup

Eseguire il backup di un server AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

È possibile definire un backup del AWS OpsWorks for Chef Automate server ricorrente giornaliero o settimanale e fare archiviare i backup in Amazon Simple Storage Service (Amazon S3) per tuo conto. In alternativa, puoi effettuare backup manuali on demand.

Poiché i backup vengono archiviati in Amazon S3, sono soggetti a costi aggiuntivi. È possibile definire un periodo di retention dei backup fino a 30 generazioni. Puoi inviare una richiesta di assistenza per modificare tale limite utilizzando i canali di supporto. AWS I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

È possibile aggiungere tag ai backup di un AWS OpsWorks for Chef Automate server. Se sono stati aggiunti tag a un server AWS OpsWorks for Chef Automate , i backup automatizzati del server ereditano tali tag. Per ulteriori informazioni su come aggiungere e gestire i tag nei backup, vedere Lavorare con i tag sulle AWS OpsWorks for Chef Automate risorse in questa guida.

Argomenti

- Backup automatici
- Backup manuali
- Eliminare i backup

Backup automatici

Quando configuri il AWS OpsWorks for Chef Automate server, scegli tra backup automatici o manuali. AWS OpsWorks for Chef Automate avvia i backup automatici nell'ora e nel giorno scelti nella sezione Backup automatico della pagina Configura impostazioni avanzate del programma di configurazione. Una volta che il server è online, puoi modificare le impostazioni di backup effettuando

i seguenti passaggi, dal riquadro del server nella home page dei server Chef Automate o nella pagina delle proprietà del server.

Per modificare le impostazioni del backup automatico

1. Nel menu Actions (Azioni) del riquadro del server nella home page Chef servers (Server Chef), scegliere Change settings (Modifica impostazioni)
2. Per disattivare i backup automatici, scegliere No per l'opzione Enable automated backups (Abilita backup automatici). Salvare le modifiche. Non è necessario procedere con il passaggio successivo.
3. Nella sezione Automated Backup (Backup automatico) modificare la frequenza, l'ora di inizio o le generazioni da conservare. Salvare le modifiche.

Backup manuali

È possibile avviare un backup manuale in qualsiasi momento in o AWS Management Console eseguendo il comando AWS CLI create-backup. I backup manuali non sono inclusi nelle 30 generazioni massime di backup automatiche archiviate; vengono archiviati un massimo di 10 backup manuali che devono essere eliminati manualmente da Amazon S3.

È possibile aggiungere tag quando si crea un nuovo backup manuale di un AWS OpsWorks for Chef Automate server. Per ulteriori informazioni su come aggiungere tag quando si crea un backup manuale, vedere Aggiungere tag a un nuovo backup (CLI).

Per eseguire un backup manuale in AWS Management Console

1. Nella pagina Chef Automate servers (Server Chef Automate) scegliere il server di cui si desidera eseguire il backup.
2. Nella pagina delle proprietà del server, nel riquadro di navigazione a sinistra, scegliere Backups (Backup).
3. Scegliere Create backup (Crea backup).
4. Il backup manuale finisce quando viene visualizzato un segno di spunta verde nella colonna Status (Stato) del backup.

Per eseguire un backup manuale in AWS CLI

- Per avviare un backup manuale, esegui il AWS CLI comando riportato di seguito.

```
aws opsworks-cm --region region name create-backup --server-name "Chef server name"
 --description "optional descriptive string"
```

Eliminare i backup

L'eliminazione di un backup comporta la sua eliminazione definitiva dal bucket S3 in cui vengono archiviati i backup.

Per eliminare un backup nella AWS Management Console

1. Nella pagina Chef Automate servers (Server Chef Automate) scegliere il server di cui si desidera eseguire il backup.
2. Nella pagina delle proprietà del server, nel riquadro di navigazione a sinistra, scegliere Backups (Backup).
3. Scegliere il backup da eliminare, quindi scegliere Delete backup (Elimina backup). Puoi selezionare un solo backup alla volta.
4. Quando viene richiesto di confermare l'eliminazione, selezionare la casella di controllo Delete the backup, which is stored in an S3 bucket (Elimina il backup archiviato in un bucket S3) e scegliere Yes, Delete (Sì, elimina).

Per eliminare un backup nella AWS CLI

- Per eliminare un backup, esegui il AWS CLI comando seguente, sostituendolo `--backup-id` con l'ID del backup che desideri eliminare. IDs I backup sono nel formato*ServerName-yyyyMMddHHmmssSSS*. Ad esempio, **`test-chef-server-20171218132604388`**.

  ```
  aws opsworks-cm --region region name delete-backup --backup-id ServerName-
  yyyyMMddHHmmssSSS
  ```

Ripristino di un AWS OpsWorks for Chef Automate server da un backup

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti

di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Dopo aver esaminato i backup disponibili, puoi scegliere un momento temporale a partire dal quale ripristinare il server. AWS OpsWorks for Chef Automate I backup del server contengono solo i dati persistenti del software di gestione della configurazione (libri di ricette, nodi registrati, ecc.). L'esecuzione di un ripristino sul posto di un server (ovvero il ripristino del AWS OpsWorks for Chef Automate server esistente su una nuova EC2 istanza) registra nuovamente i nodi registrati al momento del backup utilizzato per ripristinare il server e trasferisce il traffico alla nuova istanza se il ripristino ha esito positivo e lo stato del server ripristinato è lo stesso. AWS OpsWorks for Chef Automate `Healthy` Il ripristino su un server AWS OpsWorks for Chef Automate appena creato non gestisce le connessioni dei nodi. Il ripristino di un server non aggiorna le versioni secondarie del software Chef, ma applica le stesse versioni Chef e gli stessi dati di gestione della configurazione disponibili nel backup scelto.

Il ripristino di un server richiede in genere più tempo rispetto alla creazione di un nuovo server; il tempo dipende dalla dimensione del backup scelto. Una volta completato il ripristino, la vecchia EC2 istanza rimane nello `Stopped` stato `Running` or, ma solo temporaneamente. Alla fine viene terminata.

In questa versione, è possibile utilizzare il AWS CLI per ripristinare un server Chef in AWS OpsWorks for Chef Automate.

> **Note**
>
> Puoi eseguire inoltre il comando restore-server per modificare il tipo di istanza corrente o per ripristinare o configurare la chiave SSH qualora venga persa o compromessa.

Per ripristinare un server da un backup

1. In AWS CLI, esegui il comando seguente per restituire un elenco di backup disponibili e relativi IDs. Annotare l'ID del backup che si desidera utilizzare. IDs I backup sono nel formato*myServerName-yyyyMMddHHmmssSSS*.

   ```
   aws opsworks-cm --region region name describe-backups
   ```

2. Esegui il comando seguente.

```
aws opsworks-cm --region region name restore-server --backup-id "myServerName-
yyyyMMddHHmmssSSS" --instance-type "Type of instance" --key-pair "name of your EC2
 key pair" --server-name "name of Chef server"
```

Di seguito è riportato un esempio.

```
aws opsworks-cm --region us-west-2 restore-server --backup-id
 "MyChefServer-20161120122143125" --server-name "MyChefServer"
```

3. Attendi il completamento del ripristino.

Manutenzione del sistema in AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

La manutenzione obbligatoria del sistema garantisce che le ultime versioni minori di Chef Server e Chef Automate Server, inclusi gli aggiornamenti di sicurezza, siano sempre in esecuzione su un server. AWS OpsWorks for Chef Automate La manutenzione del sistema è richiesta almeno una volta alla settimana. Utilizzando AWS CLI, è possibile configurare la manutenzione automatica giornaliera, se lo si desidera. È inoltre possibile utilizzare il AWS CLI per eseguire la manutenzione del sistema su richiesta, oltre alla manutenzione programmata del sistema.

Quando le versioni secondarie del software Chef diventano disponibili, la manutenzione del sistema viene progettata per aggiornare automaticamente la versione secondaria di Chef Automate e Chef Server sul server, non appena supera il test AWS. AWS esegue test approfonditi per verificare che gli upgrade di Chef siano pronti per la produzione e non interferiscano con gli ambienti dei clienti esistenti, quindi possono verificarsi ritardi tra le versioni del software Chef e la loro disponibilità per l'applicazione sui server Chef Automate esistenti OpsWorks . Per aggiornare le versioni secondarie disponibili del software Chef on demand, consulta Avvio della manutenzione del sistema on demand in questo argomento.

La manutenzione del sistema avvia una nuova istanza da un backup eseguito come parte del processo di manutenzione, il che aiuta a ridurre il rischio di EC2 istanze Amazon danneggiate o danneggiate sottoposte a manutenzione periodica.

> **Important**
>
> La manutenzione del sistema elimina le configurazioni o i file personalizzati che hai aggiunto al server AWS OpsWorks for Chef Automate . Per ulteriori informazioni su come riparare la configurazione o ovviare alla perdita di file, consulta Ripristino di configurazioni e file personalizzati dopo la manutenzione in questo argomento.

Argomenti

- Garantire che i nodi si fidino dell'Autorità di certificazione AWS OpsWorks
- Configurazione della manutenzione del sistema
- Avvio della manutenzione del sistema on demand
- Ripristino di configurazioni e file personalizzati dopo la manutenzione

Garantire che i nodi si fidino dell'Autorità di certificazione AWS OpsWorks

> **Note**
>
> I passaggi descritti in questa sezione non sono necessari se si utilizza un dominio e un certificato personalizzati con il AWS OpsWorks for Chef Automate server.

I nodi che gestisci con un AWS OpsWorks for Chef Automate server devono autenticarsi con il server utilizzando i certificati. Durante la manutenzione del sistema, AWS OpsWorks sostituisce l'istanza del server e rigenera nuovi certificati tramite l'autorità di AWS OpsWorks certificazione (CA). Per ripristinare automaticamente la comunicazione con i nodi gestiti al termine della manutenzione, i nodi devono considerare attendibile la AWS OpsWorks CA fornita con lo starter kit e ospitata nelle regioni supportate da. AWS OpsWorks for Chef Automate Quando si utilizza la AWS OpsWorks CA per stabilire la fiducia tra nodi e server, i nodi si riconnettono alla nuova istanza del server dopo la manutenzione. Se si aggiungono EC2 nodi utilizzando EC2 `userdata` lo script descritto inAggiungi nodi automaticamente in AWS OpsWorks for Chef Automate, i nodi sono già configurati per considerare attendibile la AWS OpsWorks CA.

- Per i nodi basati su Linux, la posizione nel bucket S3 della CA è `https://opsworks-cm-${REGION}-prod-default-assets.s3.amazonaws.com/misc/opsworks-cm-ca-2020-root.pem`. La CA AWS OpsWorks affidabile deve essere archiviata nel percorso`/etc/chef/opsworks-cm-ca-2020-root.pem`.
- Per i nodi basati su Windows, la posizione nel bucket S3 della CA è `https://opsworks-cm-$env:AWS_REGION-prod-default-assets.s3.amazonaws.com/misc/opsworks-cm-ca-2020-root.pem`. La AWS OpsWorks CA deve essere archiviata nella cartella principale di Chef; ad esempio, `C:\chef\opsworks-cm-ca-2020-root.pem`

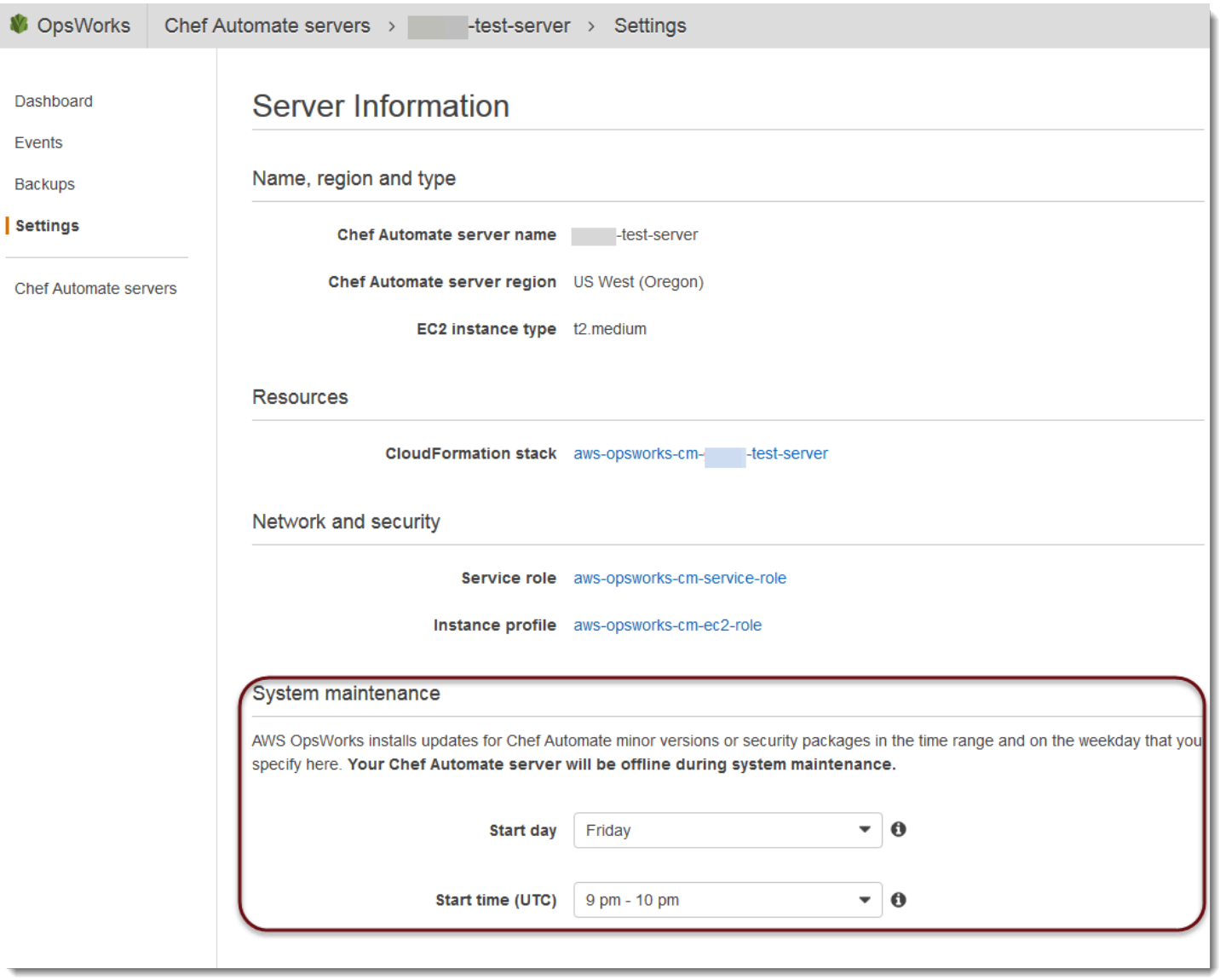
In entrambi i percorsi, la variabile relativa alla regione restituisce uno dei seguenti valori.

- `us-east-2`
- `us-east-1`
- `us-west-1`
- `us-west-2`
- `ap-northeast-1`
- `ap-southeast-1`
- `ap-southeast-2`
- `eu-central-1`
- `eu-west-1`

Configurazione della manutenzione del sistema

Quando si crea un nuovo AWS OpsWorks for Chef Automate server, è possibile configurare un giorno della settimana e un'ora, in formato UTC (Coordinated Universal Time), per l'avvio della manutenzione del sistema. La manutenzione viene avviata all'ora specificata. In quanto ci si deve aspettare che il server sia offline durante il mantenimento del sistema, scegliere un orario di utilizzo del server basso durante l'orario di ufficio regolare. Durante la manutenzione, lo stato del server è `UNDER_MAINTENANCE`.

È inoltre possibile modificare le impostazioni di manutenzione del sistema su un AWS OpsWorks for Chef Automate server esistente, modificando le impostazioni nell'area Manutenzione del sistema della pagina Impostazioni del server, come mostrato nella schermata seguente.



Nella sezione System maintenance (Manutenzione sistema), impostare il giorno e l'ora quando si desidera che il mantenimento del sistema inizi.

Configurazione della manutenzione del sistema utilizzando il AWS CLI

Puoi anche configurare l'ora di avvio automatico per la manutenzione del sistema tramite AWS CLI. AWS CLI Consente di configurare la manutenzione automatica giornaliera, se lo si desidera, omettendo il prefisso di tre caratteri del giorno della settimana.

In un comando `create-server` aggiungi il parametro `--preferred-maintenance-window` dopo aver specificato i requisiti per la creazione dell'istanza del server (ad esempio tipo di istanza, ARN del profilo dell'istanza e ARN del ruolo di servizio). Nell'esempio `create-server` seguente, `--`

`preferred-maintenance-window` è impostato su `Mon:08:00`, quindi la manutenzione è stata impostata per essere avviata ogni lunedì mattina alle 8:00 UTC.

```
aws opsworks-cm create-server --engine "Chef" --engine-model "Single" --
engine-version "12" --server-name "automate-06" --instance-profile-arn
 "arn:aws:iam::1019881987024:instance-profile/aws-opsworks-cm-ec2-role"
 --instance-type "t2.medium" --key-pair "amazon-test" --service-role-arn
 "arn:aws:iam::044726508045:role/aws-opsworks-cm-service-role" --preferred-maintenance-
window "Mon:08:00"
```

In un comando `update-server` puoi aggiornare solo il valore `--preferred-maintenance-window`, se lo desideri. Nell'esempio seguente la finestra di manutenzione è impostata su venerdì sera alle 18:15 UTC.

```
aws opsworks-cm update-server --server-name "shiny-kitchen" --preferred-maintenance-
window "Fri:18:15"
```

Per modificare l'ora di inizio della finestra di manutenzione in modo che sia impostata sulle 18:15 UTC tutti i giorni, ometti il prefisso di tre caratteri relativo al giorno feriale, come illustrato nell'esempio seguente.

```
aws opsworks-cm update-server --server-name "shiny-kitchen" --preferred-maintenance-
window "18:15"
```

Per ulteriori informazioni sull'impostazione della finestra preferita di manutenzione del sistema utilizzando AWS CLI, consulta create-server e update-server.

Avvio della manutenzione del sistema on demand

Per avviare la manutenzione del sistema su richiesta, al di fuori della manutenzione automatica settimanale o giornaliera configurata, esegui il comando seguente. AWS CLI Non è possibile avviare la manutenzione on-demand nella AWS Management Console.

```
aws opsworks-cm start-maintenance --server-name server_name
```

Per ulteriori informazioni su questo comando, consulta start-maintenance.

Ripristino di configurazioni e file personalizzati dopo la manutenzione

La manutenzione del sistema può eliminare o modificare file o configurazioni personalizzati aggiunti al AWS OpsWorks for Chef Automate server.

Se, dopo un'esecuzione di manutenzione, sul server Chef mancano i file o le impostazioni che hai aggiunto utilizzando `RunCommand` o SSH, puoi utilizzare un'Amazon Machine Image (AMI) per avviare una nuova EC2 istanza Amazon. AMIs sono disponibili basate sulla configurazione di pre-manutenzione di un server.

La nuova istanza si trova nello stesso stato del server Chef prima della manutenzione e dovrebbe includere le impostazioni e i file mancanti.

> **Important**
>
> Non è possibile utilizzare la nuova istanza per ripristinare il server. L'istanza non può essere eseguita come server Chef. Puoi utilizzare l'istanza solo per ripristinare le impostazioni di configurazione e i file.

Per avviare un' EC2 istanza da un'AMI, nella EC2 console Amazon, apri la procedura guidata di avvio, scegli Mio AMIs, quindi scegli l'AMI con il nome del tuo server. Segui i passaggi della EC2 procedura guidata di Amazon come faresti per il lancio di qualsiasi altra istanza.

Scansioni di conformità in AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Le scansioni di conformità ti permettono di monitorare la conformità dei nodi gestiti nell'infrastruttura in base a policy predefinite, chiamate anche regole. Le visualizzazioni Compliance ti permettono di controllare regolarmente le applicazioni per individuarne eventuali vulnerabilità e configurazioni non conformi. Chef offre oltre 100 profili di conformità predefiniti, ovvero raccolte di regole valide per

configurazioni di nodi specifiche, che puoi usare nelle scansioni di conformità. Puoi anche usare il InSpec linguaggio Chef per creare i tuoi profili personalizzati.

Se il server ancora non esegue Chef Automate 2.0, è possibile configurare Chef Compliance manualmente, installando il libro di ricette Audit.

> **Note**
>
> La versione minima supportata del software Chef Infra client agent (`chef-client`) sui nodi associati a un AWS OpsWorks for Chef Automate server è 13. x. Ti consigliamo di utilizzare la `chef-client` versione più recente e stabile o almeno la 14.10.9.

Argomenti

- Conformità in Chef Automate 2.0
- Confomità in Chef Automate 1.x
- Aggiornamenti di conformità
- Profili Compliance della community e personalizzati
- Vedi anche

Conformità in Chef Automate 2.0

Se sul tuo AWS OpsWorks for Chef Automate server è in esecuzione Chef Automate 2.0, configura Chef Compliance utilizzando le procedure in questa sezione.

Esecuzione di processi di scansione di conformità con Chef Automate 2.0

Chef Automate 2.0 include la funzionalità di InSpec scansione della conformità di Chef che in precedenza richiedeva la configurazione manuale e la configurazione del libro di cucina. È possibile eseguire lavori di scansione su un server che esegue Chef Automate 2.0 AWS OpsWorks for Chef Automate . Le operazioni possono essere eseguite immediatamente (una volta), pianificate per un secondo momento o pianificate per essere eseguite a intervalli specificati, ad esempio ogni giorno o ogni due ore. I risultati di un processo di scansione vengono inviati al report di conformità. È possibile visualizzare e intraprendere azioni sui risultati della scansione di conformità nel pannello di controllo di Chef Automate. Per aprire la scheda Compliance (Conformità) e visualizzare i report, nella scheda Scan Jobs (Processi di scansione) nel pannello di controllo di Chef Automate, scegli Report a destra della riga di un nodo gestito.

Per eseguire i processi di scansione sui nodi gestiti, è necessario disporre di quanto segue.

- Almeno un profilo di conformità installato nello spazio dei nomi.
- Almeno un nodo di destinazione, aggiunto manualmente o un' EC2 istanza aggiunta automaticamente.

In AWS OpsWorks for Chef Automate, i lavori di scansione sono supportati sulle seguenti destinazioni.

- Nodi aggiunti manualmente
- Istanze `aws-ec2`
- Regioni AWS

Per istruzioni dettagliate su come eseguire i processi di scansione, consulta l'argomento relativo ai processi di scansione di Chef Automate nella documentazione.

(Opzionale, Chef Automate 2.0) Impostazione della conformità con il libro di ricette Audit

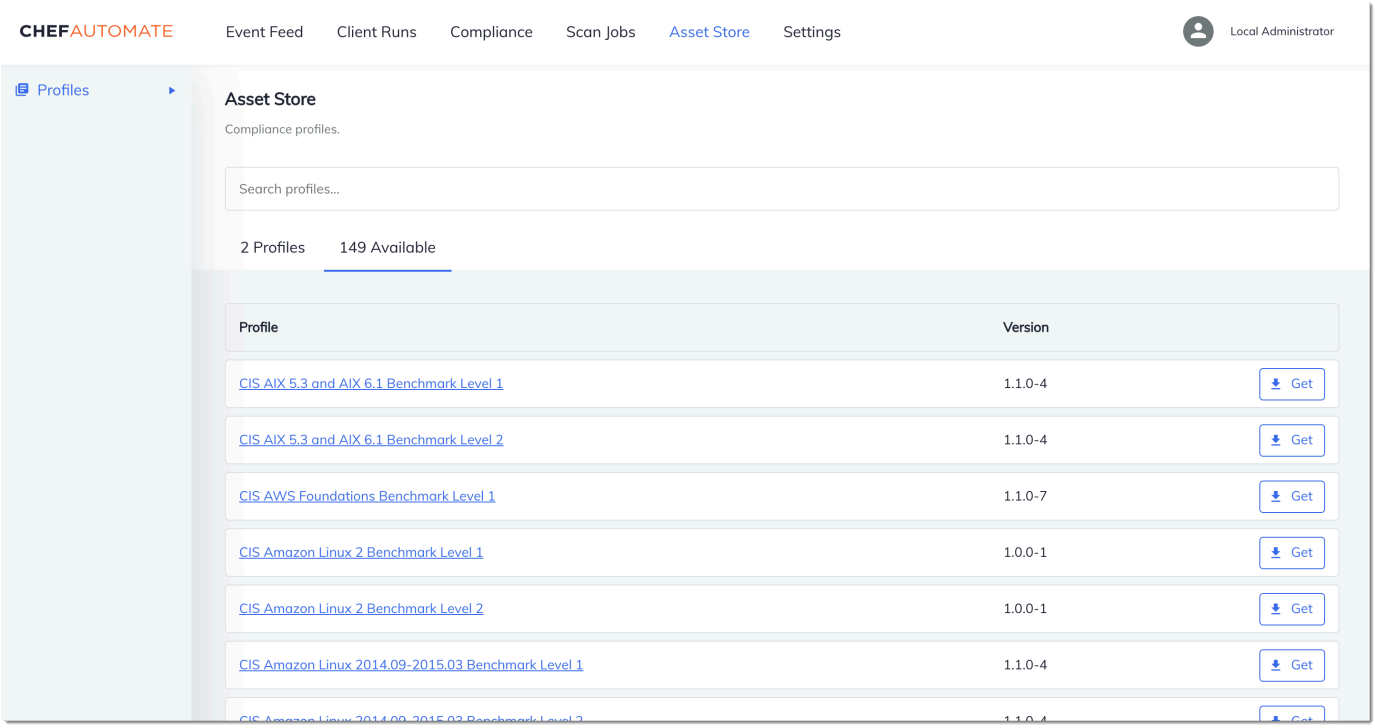
È possibile configurare la conformità su qualsiasi AWS OpsWorks for Chef Automate server. Dopo l'avvio di un server AWS OpsWorks for Chef Automate , puoi installare i profili dal pannello di controllo Chef Automate o aggiungere i profili desiderati agli attributi del libro di ricette Audit nel file di policy `Policyfile.rb`. Un file `Policyfile.rb` precompilato è incluso nello starter kit.

Dopo la modifica di `Policyfile.rb` con i profili come attributi del libro di ricette Audit, esegui i comandi `chef push` per caricare il libro di ricette Audit e altri libri di ricette specificati in `Policyfile.rb` nel server Chef Automate. L'installazione del ricettario Audit installa anche il gem for Chef InSpec, un framework di test e audit open source prodotto da Chef. Per Chef Automate 2.0, scegli la versione 7.1.0 o successiva del libro di ricette Audit. La InSpec gem deve essere la versione 2.2.102 o successiva.

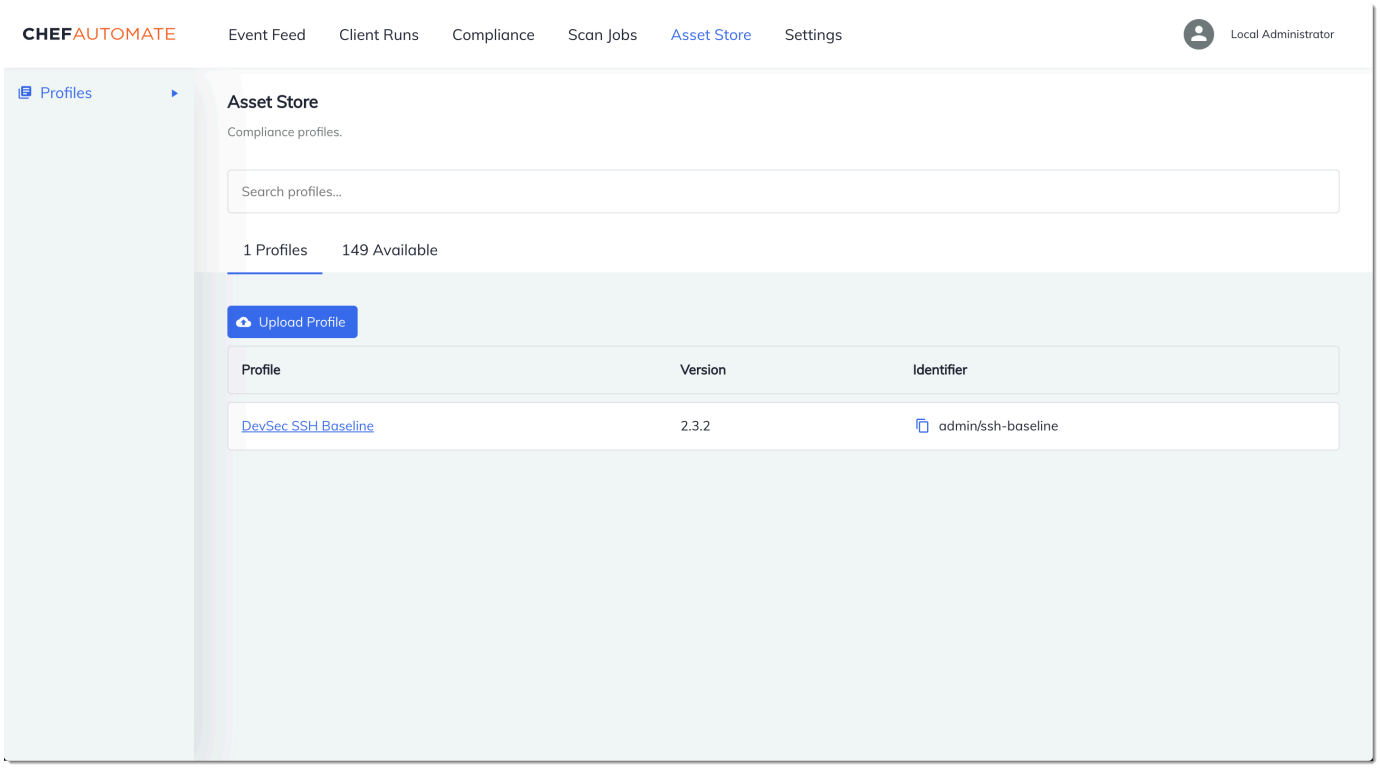
Le istruzioni incluse in questa sezione mostrano come implementare il libro di ricette `opsworks-audit`. Il ricettario Audit scarica i profili specificati dal server Chef Automate, valuta i nodi rispetto al profilo DevSec SSH Baseline e riporta il risultato delle scansioni di conformità ad ogni esecuzione. `chef-client`

Per installare profili di conformità

1. Se non è già stato fatto, accedere al pannello di controllo Web di Chef Automate. Utilizzare le credenziali ricevute al momento del download dello starter kit durante la creazione del server AWS OpsWorks for Chef Automate .
2. Nel pannello di controllo di Chef Automate, scegliere la scheda Asset Store (Store asset).



3. Scegliere la scheda Available (Disponibile) per vedere i profili predefiniti.
4. Esaminare l'elenco dei profili. Scegliere un profilo corrispondente al sistema operativo e alla configurazione di almeno uno dei nodi gestiti. Per visualizzare i dettagli sul profilo, inclusa una descrizione delle violazioni identificate e del codice delle regole sottostante del profilo, scegliere > a destra della voce del profilo. È possibile scegliere più profili. Se stai configurando l'esempio nello Starter Kit, scegli SSH Baseline. DevSec



5. Per installare i profili selezionati nel server Chef Automate, scegliere Get (Ottieni).
6. I profili una volta installati vengono visualizzati nella scheda Profiles (Profili) del pannello di controllo di Chef Automate.

Per installare i libri di ricette con **`Policyfile.rb`**

1. Visualizzare `Policyfile.rb` nello starter kit per verificare che gli attributi per il libro di ricette Audit specifichino il profilo `ssh-baseline` in `['profiles']`.

```
# Define audit cookbook attributes
default["opsworks-demo"]["audit"]["reporter"] = "chef-server-automate"
default["opsworks-demo"]["audit"]["profiles"] = [
  {
    "name": "DevSec SSH Baseline",
    "compliance": "admin/ssh-baseline"
  }
]
```

2. Scaricare e installare i libri di ricette definiti in `Policyfile.rb`.

```
chef install
```

La versione di tutti i libri di ricette è indicata nel file `metadata.rb` del libro di ricette. A ogni modifica di un libro di ricette, è necessario aumentare la versione del libro di ricette all'interno del file `metadata.rb`.

3. Eseguire il push della policy `opsworks-demo`, definita in `Policyfile.rb`, per il server.

```
chef push opsworks-demo
```

4. Verificare l'installazione della policy. Esegui il comando seguente.

```
chef show-policy
```

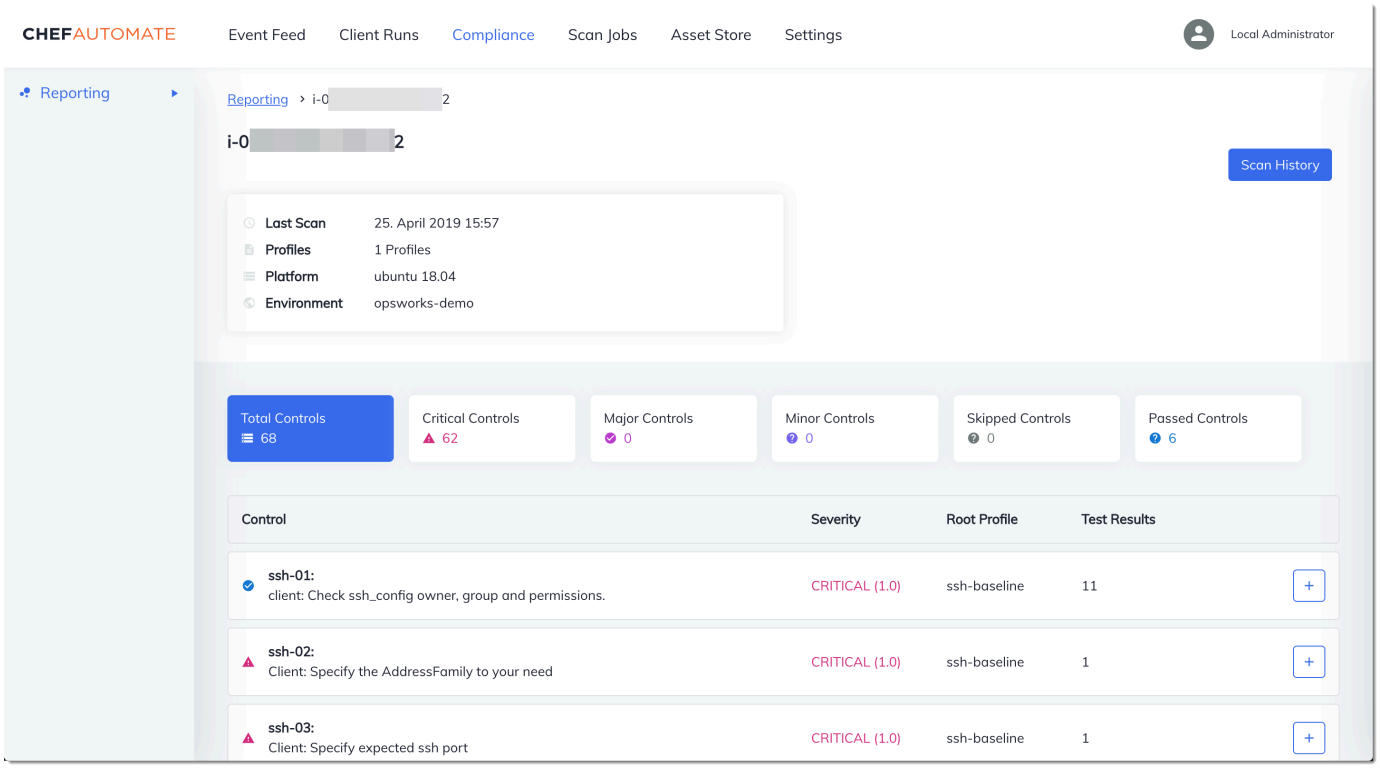
Il risultato deve essere simile al seguente:

```
opsworks-demo-webserver
=======================
* opsworks-demo:  ec0fe46314
```

5. Aggiungere i nodi al server da gestire, se non è già stato fatto. Per connettere il primo nodo al AWS OpsWorks for Chef Automate server, utilizzate lo `userdata.sh` script incluso in questo Starter Kit. Utilizza l' AWS OpsWorks `AssociateNode`API per connettere un nodo al tuo server.

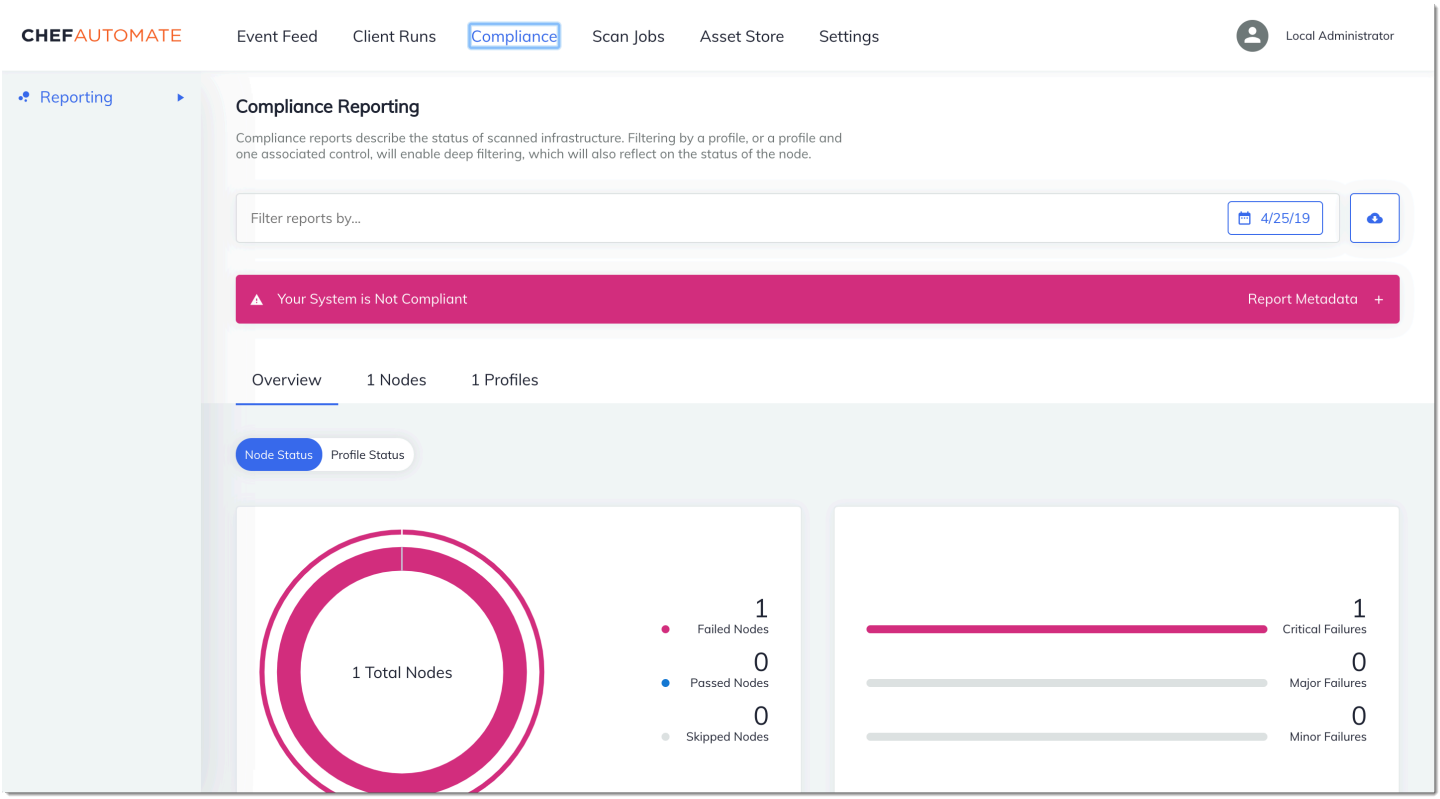
   È possibile automatizzare l'associazione di nodi completando la procedura in Aggiungi nodi automaticamente in AWS OpsWorks for Chef Automate o aggiungendo un nodo alla volta completando la procedura in Aggiungi nodi singolarmente.

6. Dopo aver aggiornato l'elenco di esecuzione per i nodi, l'agente `chef-client` esegue le ricette specificate alla successiva esecuzione. Per impostazione predefinita, l'esecuzione avviene ogni 1800 secondi (30 minuti). Dopo l'esecuzione, è possibile visualizzare e intraprendere azioni sui risultati di conformità dalla scheda Compliance (Conformità) nel pannello di controllo di Chef Automate.

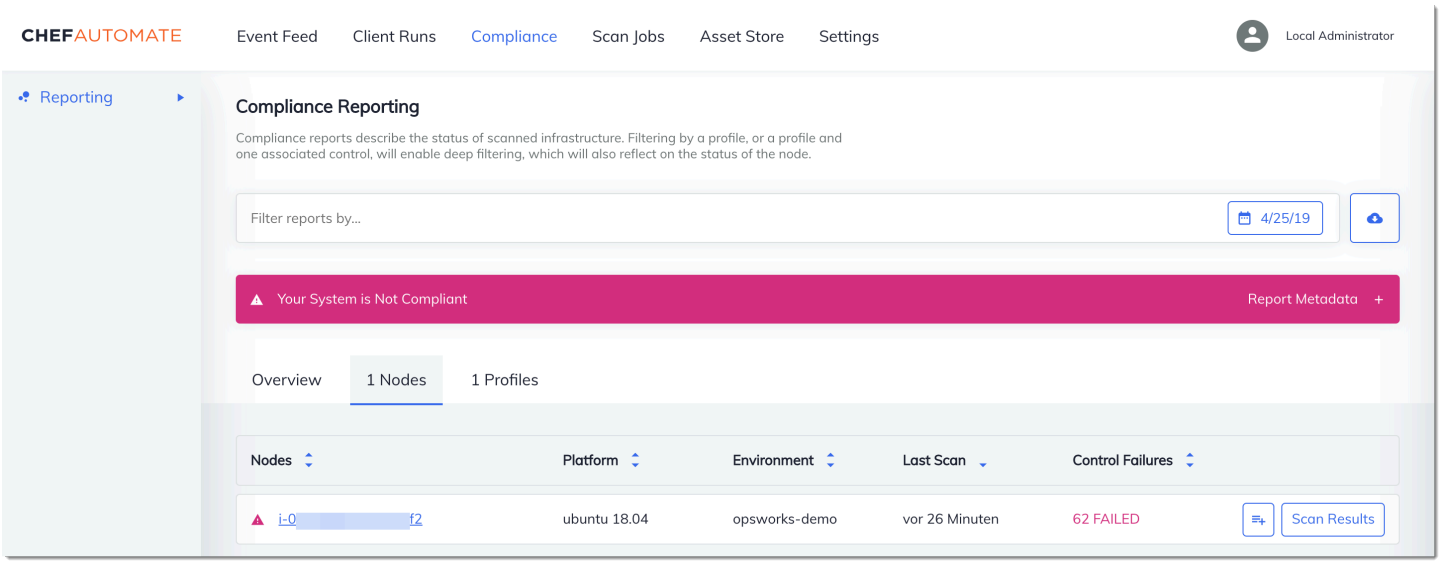


Esecuzione di una scansione di compliance

I risultati della scansione di compliance vengono in genere visualizzati nel pannello di controllo di Chef Automate subito dopo la prima esecuzione dell'agente, che avviene dopo la configurazione degli elenchi di esecuzione dei nodi.



Nel pannello di controllo di Chef Automate scegliere la scheda Compliance. Nel riquadro di navigazione a sinistra, scegliere Reporting (Creazione di report). Scegliere la scheda Profiles (Profili), scegliere Scan Results (Risultati scansione) e quindi scegliere un nodo con errori di scansione per ottenere altre informazioni sulle regole rispetto alle quali il nodo ha restituito un errore.



In genere, vengono visualizzati risultati di scansione non conformi, poiché i nuovi nodi non soddisfano ancora tutte le regole del profilo DevSec SSH Baseline. L'DevSec Hardening Framework, un progetto

basato sulla comunità, offre libri di cucina per risolvere problemi che violano le regole del profilo SSH Baseline. DevSec

(Facoltativo) Risoluzione dei risultati non conformi

Lo starter kit include un ricettario open source che puoi eseguire per correggere i risultati non conformi delle **`ssh-hardening`** esecuzioni con il profilo SSH Baseline. DevSec

> **Note**
>
> Il **`ssh-hardening`** ricettario apporta modifiche ai nodi per conformarsi alle regole SSH Baseline. DevSec Prima di eseguire questo libro di cucina su qualsiasi nodo di produzione, consulta i dettagli sul profilo DevSec SSH Baseline nella console Chef Automate per comprendere le violazioni delle regole a cui il libro di cucina mira. Esamina le informazioni sul libro di ricette `ssh-hardening` open source prima di eseguirlo in qualsiasi nodo di produzione.

Per eseguire il libro di ricette **`ssh-hardening`**

1. In un editor di testo, accodare il libro di ricette `ssh-hardening` all'elenco di esecuzioni di `Policyfile.rb`. L'elenco di esecuzioni di `Policyfile.rb` deve corrispondere a quanto riportato di seguito.

   ```
   run_list  'chef-client', 'opsworks-webserver', 'audit', 'ssh-hardening'
   ```

2. Aggiornare `Policyfile.rb` ed eseguirne il push al server AWS OpsWorks for Chef Automate .

   ```
   chef update Policyfile.rb
      chef push opsworks-demo
   ```

3. I nodi che sono associati alla policy `opsworks-demo` aggiornano automaticamente l'elenco di esecuzioni e applicano il libro di ricette `ssh-hardening` alla successiva esecuzione di `chef-client`.

   Poiché si utilizza il libro di ricette `chef-client`, il nodo esegue l'archiviazione a intervalli regolari (30 minuti per impostazione predefinita). Al check-in successivo, il **`ssh-hardening`** ricettario viene eseguito e aiuta a migliorare la sicurezza dei nodi per soddisfare le regole del DevSec profilo SSH Baseline.

4. Dopo l'esecuzione iniziale del libro di ricette `ssh-hardening`, attendere 30 minuti per una nuova esecuzione di una scansione di conformità. Visualizzare i risultati nel pannello di controllo di Chef Automate. I risultati non conformi che si sono verificati nell'esecuzione iniziale della scansione DevSec SSH Baseline devono essere risolti.

Confomità in Chef Automate 1.x

Se sul tuo AWS OpsWorks for Chef Automate server è in esecuzione Chef Automate 1. x, configura Chef Compliance utilizzando le procedure in questa sezione.

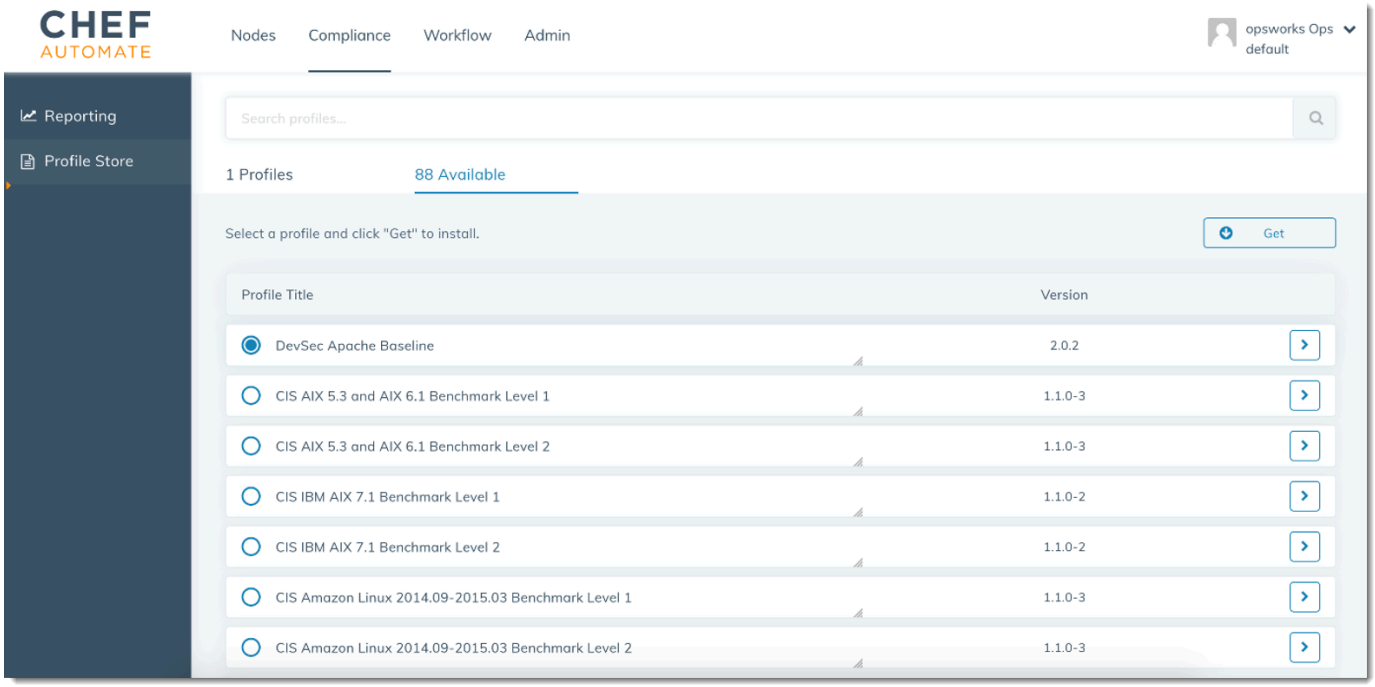
(Opzionale, Chef Automatizzare 1.x) Impostazione di Chef Compliance

Puoi configurare Chef Compliance su qualsiasi AWS OpsWorks for Chef Automate server. Dopo aver avviato un server AWS OpsWorks for Chef Automate , scegli i profili che vuoi eseguire tra quelli inclusi nel pannello di controllo di Chef Automate. Dopo aver installato i profili, esegui i comandi `berks` per caricare il libro di ricette Audit nel server Chef Automate. L'installazione del ricettario Audit installa anche gem for InSpec, un framework di test open source prodotto da Chef che consente di integrare test automatizzati in qualsiasi fase della pipeline di distribuzione. Per Chef Automate 1.x, scegli la versione 5.0.1 o successiva del libro di ricette Audit. La InSpec gem deve essere la versione 1.24.0 o successiva.

Lo AWS OpsWorks for Chef Automate starter kit include un ricettario wrapper`opsworks-audit`, che scarica e installa la versione giusta del ricettario Chef's Audit per te. Il `opsworks-audit` ricettario indica inoltre all'`chef-client`agente di valutare i nodi in base al profilo DevSecSSH Baseline che installerai dalla console di Compliance di Chef più avanti in questo argomento. Puoi configurare Compliance usando qualsiasi libro di ricette in base alle tue preferenze. Le istruzioni incluse in questa sezione mostrano come implementare il libro di ricette `opsworks-audit`.

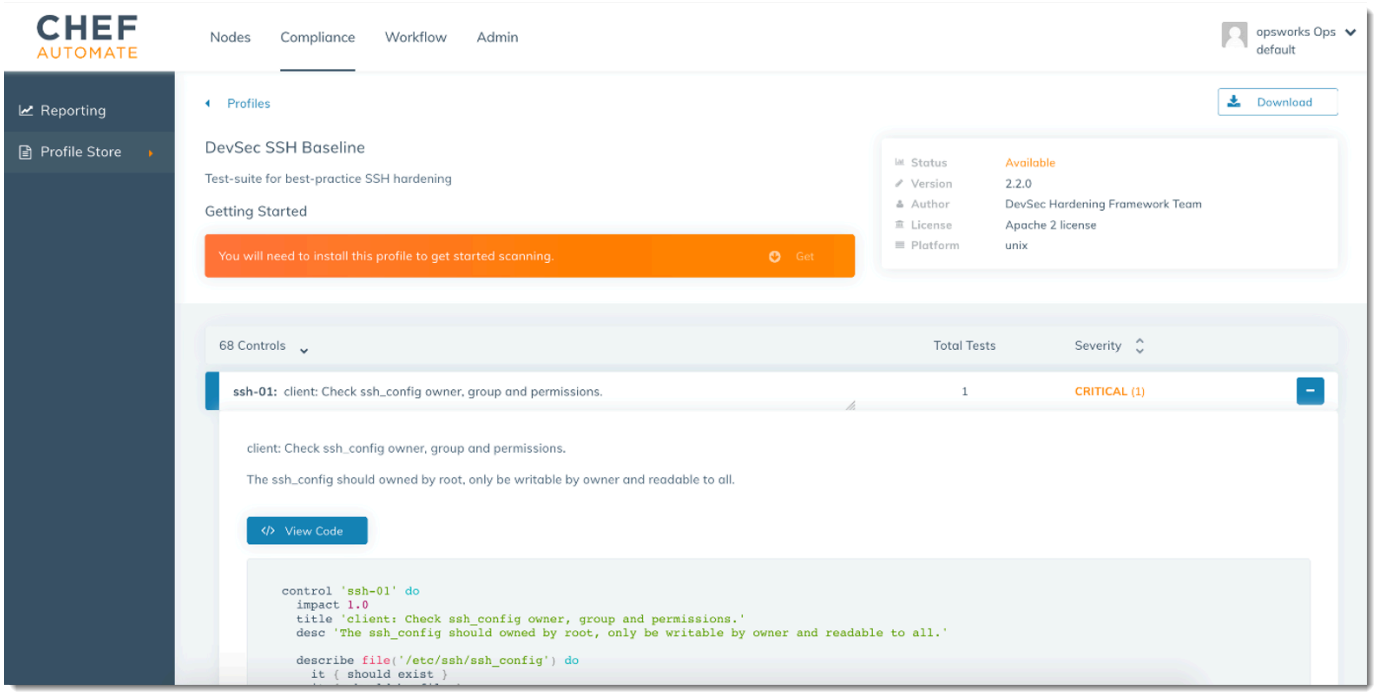
Per installare profili Compliance

1. Se non è già stato fatto, accedere al pannello di controllo Web di Chef Automate. Utilizza le credenziali che hai ricevuto quando hai scaricato lo Starter Kit durante la creazione del server. AWS OpsWorks for Chef Automate
2. Nel pannello di controllo di Chef Automate scegliere la scheda Compliance.



3. Sulla barra di navigazione a sinistra scegliere Profile Store (Store profili)e quindi scegliere la scheda Available (Disponibili) per visualizzare i profili predefiniti.

4. Esaminare l'elenco dei profili. Scegliere un profilo corrispondente al sistema operativo e alla configurazione di almeno uno dei nodi gestiti. Per visualizzare i dettagli sul profilo, inclusa una descrizione delle violazioni identificate e del codice delle regole sottostante del profilo, scegliere **>** a destra della voce del profilo. È possibile scegliere più profili.



5. Per installare i profili selezionati nel server Chef Automate, scegliere Get (Ottieni).
6. Al termine del download, passare alla procedura seguente.

Per installare e configurare il libro di ricette **`opsworks-audit`**

1. Questa fase è facoltativa, ma permette di risparmiare tempo nella fase 6, quando verranno aggiunte ricette a elenchi di esecuzione dei nodi. Modificare il file `roles/opsworks-example-role.rb` incluso nello starter kit scaricato al momento del download durante la creazione del server AWS OpsWorks for Chef Automate . Aggiungere le righe seguenti. L'ultima riga è impostata come commento, perché l'aggiunta del libro di ricette e della ricetta `ssh-hardening` per risolvere nodi non conformi dopo l'esecuzione delle scansioni Compliance è facoltativa.

   ```
   run_list(
        "recipe[chef-client]",
        "recipe[apache2]",
        "recipe[opsworks-audit]"
        # "recipe[ssh-hardening]"
          )
   ```

2. Utilizzare un editor di testo per specificare i libri di ricette desiderati nel file Berksfile. Un file Berksfile di esempio è incluso nello starter kit. In questo esempio vengono installati i libri di ricette `chef-client`, `apache2` e `opsworks-audit` del client Chef Infra. Il file Berksfile dovrebbe essere simile al seguente.

   ```
   source 'https://supermarket.chef.io
        cookbook 'chef-client'
        cookbook 'apache2', '~> 5.0.1'
        cookbook 'opsworks-audit', path: 'cookbooks/opsworks-audit', '~> 1.0.0'
   ```

   La versione di tutti i libri di ricette è indicata nel file `metadata.rb` del libro di ricette. A ogni modifica di un libro di ricette, è necessario aumentare la versione del libro di ricette all'interno del file `metadata.rb`.

3. Eseguire il comando seguente per scaricare e installare il libro di ricette nella cartella `cookbooks` nel computer locale o in uso.

   ```
   berks vendor cookbooks
   ```

4. Eseguire il comando seguente per caricare i libri di ricette gestiti da un fornitore nel server AWS OpsWorks for Chef Automate .

```
knife upload .
```

5. Eseguire il comando seguente per verificare l'installazione del libro di ricette `opsworks-audit` visualizzando un elenco di libri di ricette attualmente disponibili nel server.

```
knife cookbook list
```

6. Aggiungere i nodi al server da gestire, se non è già stato fatto. È possibile automatizzare l'associazione di nodi completando la procedura in Aggiungi nodi automaticamente in AWS OpsWorks for Chef Automate o aggiungendo un nodo alla volta completando la procedura in Aggiungi nodi singolarmente. Modificare l'elenco di esecuzione dei nodi in modo da aggiungere il ruolo specificato nella fase 1, ovvero `opsworks-example-role`. In questo esempio viene modificato l'attributo `RUN_LIST` nello script `userdata` usato per automatizzare l'associazione di nodi.

```
RUN_LIST="role[opsworks-example-role]"
```

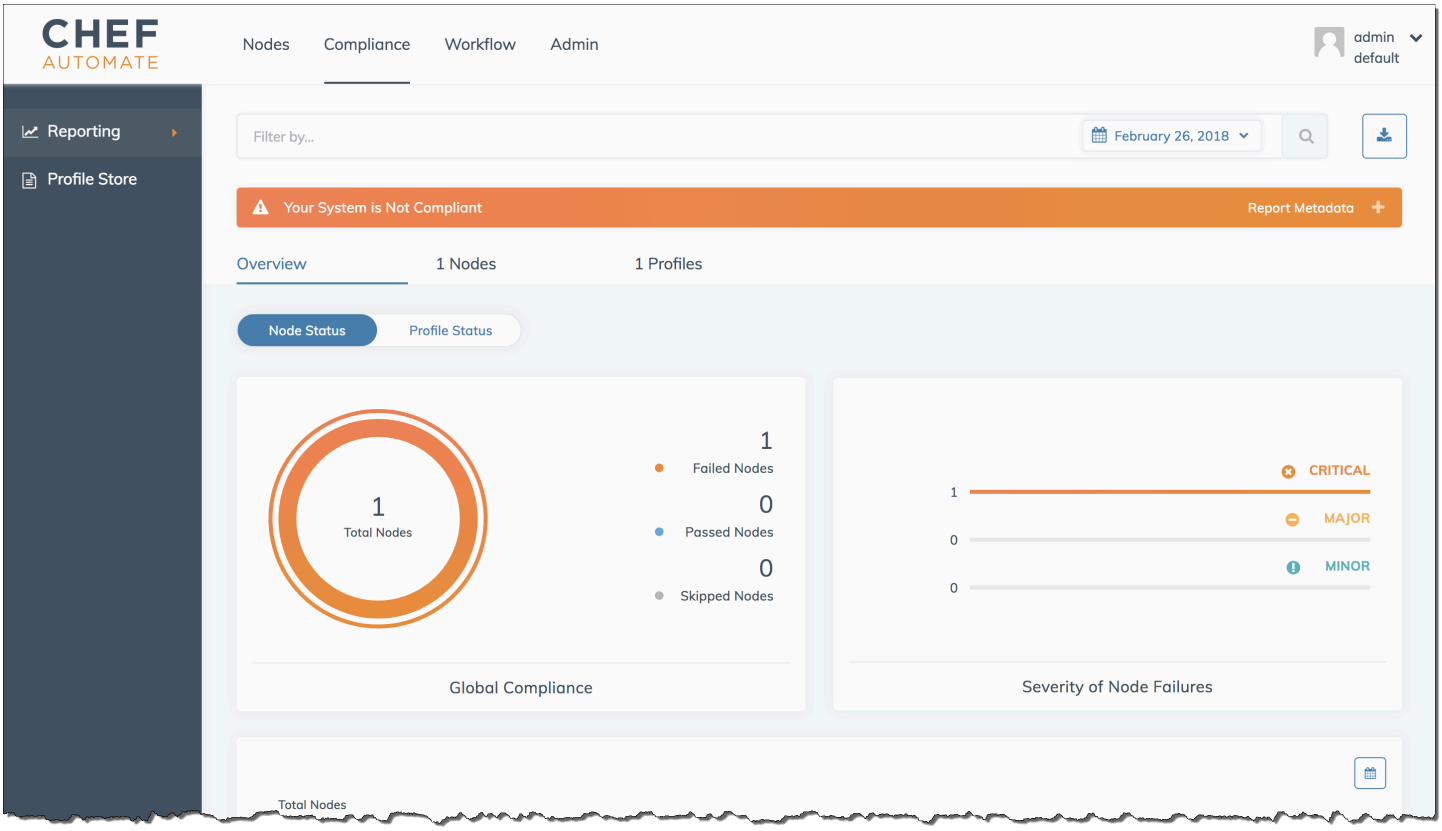
   Se la fase 1 non è stata eseguita e non è stato configurato il ruolo, aggiungere i nomi delle singole ricette all'elenco di esecuzione. Salva le modifiche e segui i passaggi Fase 3. Creare istanze tramite uno script di associazione automatica per applicare lo script userdata alle EC2 istanze Amazon.

```
RUN_LIST="recipe[chef-client],recipe[apache2],recipe[opworks-audit]"
```

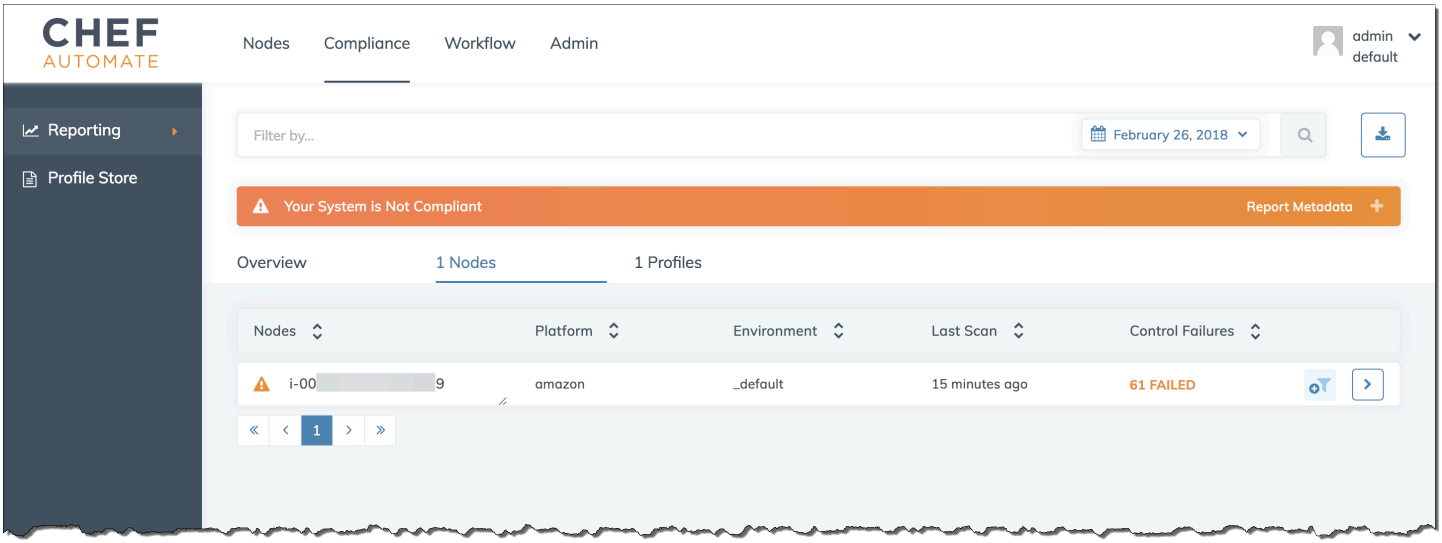
7. Dopo aver aggiornato l'elenco di esecuzione per i nodi, l'agente `chef-client` esegue le ricette specificate alla successiva esecuzione. Per impostazione predefinita, l'esecuzione avviene ogni 1800 secondi (30 minuti). Dopo l'esecuzione, i risultati di Compliance sono visibili nel pannello di controllo di Chef Automate.

Esecuzione di una scansione di compliance

I risultati della scansione di compliance vengono in genere visualizzati nel pannello di controllo di Chef Automate subito dopo la prima esecuzione del daemon dell'agente, che avviene dopo la configurazione degli elenchi di esecuzione dei nodi.



Nel pannello di controllo di Chef Automate scegliere la scheda Compliance. Nel riquadro di navigazione a sinistra, scegliere Reporting (Creazione di report). Scegliere la scheda Profiles (Profili), scegliere Scan Results (Risultati scansione) e quindi scegliere un nodo con errori di scansione per ottenere altre informazioni sulle regole rispetto alle quali il nodo ha restituito un errore.



In genere, vengono visualizzati risultati di scansione non conformi, poiché i nuovi nodi non soddisfano ancora tutte le regole del profilo SSH Baseline. DevSec L'DevSec Hardening Framework, un

progetto basato sulla comunità, offre libri di cucina per risolvere problemi che violano le regole del profilo SSH Baseline. DevSec

(Facoltativo) Risoluzione dei risultati non conformi

Lo starter kit include un ricettario open source che puoi eseguire per correggere i risultati non conformi delle **`ssh-hardening`** esecuzioni con il profilo SSH Baseline. DevSec

> **Note**
>
> Il **`ssh-hardening`** ricettario apporta modifiche ai nodi per conformarsi alle regole SSH Baseline. DevSec Prima di eseguire questo libro di cucina su qualsiasi nodo di produzione, consulta i dettagli sul profilo DevSec SSH Baseline nella console Chef Automate per comprendere le violazioni delle regole a cui il libro di cucina mira. Esamina le informazioni sul libro di ricette `ssh-hardening` open source prima di eseguirlo in qualsiasi nodo di produzione.

Per eseguire il libro di ricette **`ssh-hardening`**

1. In un editor di testo aggiungere il libro di ricette `ssh-hardening` al file Berksfile. Il file Berksfile dovrebbe essere simile al seguente.

   ```
   source 'https://supermarket.chef.io'
        cookbook 'chef-client'
        cookbook 'apache2', '~> 5.0.1'
        cookbook 'opsworks-audit', path: 'cookbooks/opsworks-audit', '~> 1.0.0' #
    optional
        cookbook 'ssh-hardening'
   ```

2. Eseguire i comandi seguenti per scaricare il libro di ricette `ssh-hardening` nella cartella dei libri di ricette locale e quindi caricarlo nel server AWS OpsWorks for Chef Automate .

   ```
   berks vendor cookbooks
   knife upload .
   ```

3. Aggiungere la ricetta `ssh-hardening` all'elenco di esecuzione dei nodi come descritto nelle fasi 1 e 6 di Per installare e configurare il libro di ricette `opsworks-audit`.

Se il file `opsworks-example-role.rb` viene aggiornato, caricare le modifiche nel server eseguendo il comando seguente.

```
knife upload .
```

Se si aggiorna direttamente l'elenco di esecuzione, caricare le modifiche eseguendo il comando seguente. Il nome del nodo è in genere l'ID istanza.

```
knife node run_list add <node name> 'recipe[ssh-hardening]'
```

4. Poiché si utilizza il libro di ricette `chef-client`, il nodo esegue l'archiviazione a intervalli regolari (30 minuti per impostazione predefinita). Al check-in successivo, il **`ssh-hardening`** ricettario viene eseguito e aiuta a migliorare la sicurezza dei nodi per soddisfare le regole del DevSec profilo SSH Baseline.
5. Dopo l'esecuzione iniziale del libro di ricette `ssh-hardening`, attendere 30 minuti per una nuova esecuzione di una scansione Compliance. Visualizzare i risultati nel pannello di controllo di Chef Automate. I risultati non conformi che si sono verificati nell'esecuzione iniziale della scansione DevSec SSH Baseline devono essere risolti.

Aggiornamenti di conformità

Su un AWS OpsWorks for Chef Automate server, la funzionalità di conformità viene aggiornata automaticamente mediante la manutenzione programmata del sistema. Man mano che le versioni aggiornate di Chef Automate, Chef Infra Server e Chef InSpec diventano disponibili per il tuo AWS OpsWorks for Chef Automate server, potresti dover controllare e aggiornare le versioni supportate di Audit cookbook e Chef InSpec gem in esecuzione sul tuo server. I profili che avete già installato sul vostro AWS OpsWorks for Chef Automate server non vengono aggiornati come parte della manutenzione.

Profili Compliance della community e personalizzati

Chef include attualmente oltre 100 profili di scansione di conformità. Puoi aggiungere profili personalizzati e della community all'elenco e quindi scaricare ed eseguire scansioni di conformità in base a tali profili, proprio come faresti per i profili inclusi. I profili di conformità basati sulla community sono disponibili in Chef Supermarket. I profili personalizzati sono programmi basati su Ruby che includono una cartella di controlli che specificano le regole di scansione.

Vedi anche

- Post di blog sull'annuncio di Chef Compliance
- Formazione online su Compliance di Chef Automate
- InSpecSito web di Chef
- InSpec Tutorial per chef

Dissociare un nodo da un server AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Questa sezione descrive come dissociare o rimuovere un nodo gestito dalla gestione da parte di un server. AWS OpsWorks for Chef Automate Questa operazione viene eseguita sulla riga di comando; non è possibile dissociare i nodi nella console di AWS OpsWorks for Chef Automate gestione. Attualmente, l' AWS OpsWorks for Chef Automate API non consente la rimozione in batch di più nodi. Il comando descritto in questa sezione consente di eliminare l'associazione di un nodo alla volta.

Consigliamo di eliminare l'associazione dei nodi da un server Chef prima di eliminare il server. In questo modo i nodi continueranno a funzionare senza tentare di riconnettersi al server. Per fare ciò, esegui il `disassociate-node` AWS CLI comando.

Per disassociare i nodi

1. In AWS CLI, esegui il comando seguente per dissociare i nodi. *`Node_name`*è il nome del nodo da cui desideri dissociare; per EC2 le istanze Amazon, questo è l'ID dell'istanza. *`Server_name`*è il nome del server Chef da cui desideri dissociare il nodo. `--engine-attributes`specifica il tuo nome predefinito`CHEF_AUTOMATE_ORGANIZATION`. Tutti e tre questi parametri sono obbligatori.

   Il parametro `--region` non è obbligatorio a meno che non si desidera dissociare un nodo da un server Chef che non si trova nella regione predefinita.

```
aws opsworks-cm --region Region_name disassociate-node --node-
name Node_name --server-name Server_name --engine-attributes
 "Name=CHEF_AUTOMATE_ORGANIZATION,Value='default'"
```

Il comando seguente è un esempio.

```
aws opsworks-cm --region us-west-2 disassociate-node --node-name
 i-0010zzz00d66zzz90 --server-name opsworkstest --engine-attributes
 "Name=CHEF_AUTOMATE_ORGANIZATION,Value='default'"
```

2. Attendere un messaggio di risposta in cui viene indicato che la disassociazione è terminata.

   Dopo aver dissociato con successo un nodo da un AWS OpsWorks for Chef Automate server, potrebbe essere ancora visibile nella dashboard di Chef Automate. Per impostazione predefinita, Chef utilizza un periodo di conservazione delle informazioni sullo stato del nodo e rimuove il nodo automaticamente dopo alcuni giorni.

Per ulteriori informazioni su come eliminare un AWS OpsWorks for Chef Automate server, consulta. Eliminare un AWS OpsWorks for Chef Automate server

Argomenti correlati

I seguenti post del AWS blog offrono ulteriori informazioni sull'associazione automatica dei nodi al server Chef Automate, utilizzando i gruppi di Auto Scaling o all'interno di più account.

- Utilizzo di AWS OpsWorks for Chef Automate per gestire EC2 istanze con Auto Scaling
- OpsWorks per Chef Automate: avvio automatico dei nodi in diversi account

Eliminare un AWS OpsWorks for Chef Automate server

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Questa sezione descrive come eliminare un AWS OpsWorks for Chef Automate server. L'eliminazione di un server comporta anche l'eliminazione dei relativi eventi, log e qualsiasi libro di ricette archiviati nel server. Vengono eliminate anche le risorse di supporto (istanza Amazon Elastic Compute Cloud, volume Amazon Elastic Block Store, ecc.), insieme a tutti i backup automatici.

Sebbene l'eliminazione di un server non comporti l'eliminazione dei nodi, questi ultimi non verranno più gestiti dal server eliminato e tenteranno continuamente di riconnettersi. Per questo motivo, ti consigliamo di disassociare i nodi gestiti prima di eliminare un server Chef. In questa versione, puoi dissociare i nodi eseguendo un comando. AWS CLI

Passaggio 1: Disassociare i nodi gestiti

Dissocia i nodi dal server Chef prima di eliminare il server. In questo modo i nodi continueranno a funzionare senza tentare di riconnettersi al server. A tale scopo, esegui il `disassociate-node` AWS CLI comando.

Per disassociare i nodi

1. In AWS CLI, esegui il comando seguente per dissociare i nodi. *`Server_name`*è il nome del server Chef da cui si desidera dissociare il nodo.

   ```
   aws opsworks-cm --region Region_name disassociate-node --node-name Node_name --
   server-name Server_name
   ```

2. Attendere un messaggio di risposta in cui viene indicato che la disassociazione è terminata.

Passaggio 2: Eliminare il server

1. Nel riquadro del server nel pannello di controllo, espandere il menu Actions (Azioni).
2. Scegliere Delete server (Elimina server).
3. Quando viene richiesto di confermare l'eliminazione, scegliere Yes (Sì).

Reimpostazione delle credenziali del pannello di controllo di Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Periodicamente potrebbe essere necessario modificare la password con cui accedi al pannello di controllo di Chef Automate. Puoi anche utilizzare i AWS CLI comandi di Amazon EC2 Systems Manager mostrati in questa sezione per modificare la password del pannello di controllo di Chef Automate se l'hai persa. Il comando che usi dipende dal fatto che sul tuo server Chef Automate sia in esecuzione la versione 1 o la versione 2 di Chef Automate.

1. Per restituire l'ID di istanza del tuo server Chef, apri AWS Management Console la pagina seguente.

   https://console.aws.amazon.com/ec2/v2/home? region= #istanze:search= *region_of_your_server* - aws-opsworks-cm *server_name*

   Ad esempio, per un server Chef denominato MyChefServernella regione Stati Uniti occidentali (Oregon), l'URL della console sarebbe il seguente.

   https://console.aws.amazon.com/ec2/v2/home? region=us-west-2 #istanze:ricerca= - aws-opsworks-cm MyChefServer

   Annotare l'ID istanza visualizzato nella console; sarà necessario per modificare la password.

2. Per reimpostare la password di accesso alla dashboard di Chef Automate, esegui uno dei seguenti AWS CLI comandi, a seconda che sul server sia in esecuzione Chef Automate 1 o Chef Automate 2. Sostituiscila *enterprise_name* con il nome della tua azienda o organizzazione, *user_name* con il nome utente di un amministratore sul server, *new_password* con la password che desideri utilizzare e *region_name* con la regione in cui si trova il server. Se non si specifica un nome di azienda, verrà utilizzato `default`. Per impostazione predefinita, *enterprise_name* è `default` (questo è il nome dell'organizzazione a cui viene sempre

assegnato il provisioning). Per*user_name*, crea AWS OpsWorks for Chef Automate solo un utente denominato`admin`. Annotare la nuova password e conservarla in una posizione sicura e comoda.

Per Chef Automate 1:

```
aws ssm send-command --document-name "AWS-RunShellScript" --comment "reset admin
 password" --instance-ids "instance_id"
--parameters commands="sudo delivery-ctl reset-
password enterprise_name user_name new_password" --region region_name --output text
```

Per Chef Automate 2:

```
aws ssm send-command --document-name "AWS-RunShellScript" --comment "reset admin
 password" --instance-ids "instance_id"
--parameters commands="sudo chef-automate iam admin-access restore new_password" --
region region_name --output text
```

3. Attendere che nel testo di output (in questo caso, l'ID comando) venga confermata l'avvenuta modifica della password.

Registrazione delle chiamate AWS OpsWorks for Chef Automate API con AWS CloudTrail

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

AWS OpsWorks for Chef Automate è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un'identità IAM o da un AWS servizio in. AWS OpsWorks for Chef Automate CloudTrail acquisisce tutte le chiamate API relative agli eventi AWS OpsWorks for Chef Automate as, incluse le chiamate dalla AWS OpsWorks for Chef Automate console e le chiamate in codice a. AWS OpsWorks for Chef Automate APIs Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per. AWS OpsWorks for

Chef Automate Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare a quale richiesta è stata inviata AWS OpsWorks for Chef Automate, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

AWS OpsWorks for Chef Automate Informazioni in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in AWS OpsWorks for Chef Automate, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi di AWS OpsWorks for Chef Automate, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un trail nella console, il trail sarà valido in tutte le regioni. Il trail registra gli eventi da tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consultare:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte AWS OpsWorks for Chef Automate le azioni vengono registrate CloudTrail e documentate nell'AWS OpsWorks for Chef Automate API Reference. Ad esempio, le chiamate a CreateServerCreateBackup, e DescribeServersle azioni generano voci nei file di CloudTrail registro.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali dell'utente IAM o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.

- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Comprensione delle AWS OpsWorks for Chef Automate voci dei file di registro

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro per l' AWS OpsWorks for Chef Automate `CreateServer`azione.

```
{"eventVersion":"1.05",
"userIdentity":{
    "type":"AssumedRole",
    "principalId":"ID number:OpsWorksCMUser",
    "arn":"arn:aws:sts::831000000000:assumed-role/Admin/OpsWorksCMUser",
    "accountId":"831000000000","accessKeyId":"ID number",
    "sessionContext":{
        "attributes":{
            "mfaAuthenticated":"false",
            "creationDate":"2017-01-05T22:03:47Z"
            },
        "sessionIssuer":{
            "type":"Role",
            "principalId":"ID number",
            "arn":"arn:aws:iam::831000000000:role/Admin",
            "accountId":"831000000000",
            "userName":"Admin"
            }
        }
    },
"eventTime":"2017-01-05T22:18:23Z",
"eventSource":"opsworks-cm.amazonaws.com",
"eventName":"CreateServer",
"awsRegion":"us-west-2",
```

```
"sourceIPAddress":"101.25.190.51",
"userAgent":"console.amazonaws.com",
"requestParameters":{
    "serverName":"OpsChef-test-server",
    "engineModel":"Single",
    "engine":"Chef",
    "instanceProfileArn":"arn:aws:iam::831000000000:instance-profile/aws-opsworks-cm-
ec2-role",
    "backupRetentionCount":3,"serviceRoleArn":"arn:aws:iam::831000000000:role/service-
role/aws-opsworks-cm-service-role",
    "engineVersion":"12",
    "preferredMaintenanceWindow":"Fri:21:00",
    "instanceType":"t2.medium",
    "subnetIds":["subnet-1e111f11"],
    "preferredBackupWindow":"Wed:08:00"
    },
"responseElements":{
    "server":{
        "endpoint":"OpsChef-test-server-thohsgreckcnwgz3.us-west-2.opsworks-cm.io",
        "createdAt":"Jan 5, 2017 10:18:22 PM",
        "serviceRoleArn":"arn:aws:iam::831000000000:role/service-role/aws-opsworks-cm-
service-role",
        "preferredBackupWindow":"Wed:08:00",
        "status":"CREATING",
        "subnetIds":["subnet-1e111f11"],
        "engine":"Chef",
        "instanceType":"t2.medium",
        "serverName":"OpsChef-test-server",
        "serverArn":"arn:aws:opsworks-cm:us-west-2:831000000000:server/OpsChef-test-
server/8epp7f6z-e91f-4z10-89z5-8c6219cdb09f",
        "engineModel":"Single",
        "backupRetentionCount":3,
        "engineAttributes":[
            {"name":"CHEF_STARTER_KIT","value":"*** Redacted ***"},
            {"name":"CHEF_PIVOTAL_KEY","value":"*** Redacted ***"},
            {"name":"CHEF_DELIVERY_ADMIN_PASSWORD","value":"*** Redacted ***"}],
        "engineVersion":"12.11.1",
        "instanceProfileArn":"arn:aws:iam::831000000000:instance-profile/aws-opsworks-
cm-ec2-role",
        "preferredMaintenanceWindow":"Fri:21:00"
        }
    },
"requestID":"de7f64f9-d394-12ug-8081-7bb0386fbcb6",
"eventID":"8r7b18df-6c90-47be-87cf-e8346428cfc3",
```

```
"eventType":"AwsApiCall",
"recipientAccountId":"831000000000"
}
```

Risoluzione dei problemi AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate ha raggiunto la fine del ciclo di vita il 5 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo ai clienti esistenti di migrare a Chef SaaS o a una soluzione alternativa. Se hai domande, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Questo argomento contiene alcuni AWS OpsWorks for Chef Automate problemi comuni e le relative soluzioni suggerite.

Argomenti

- Suggerimenti generali per la risoluzione dei problemi
- Risoluzione di errori specifici
- Supporto e informazioni della guida aggiuntivi

Suggerimenti generali per la risoluzione dei problemi

Se non sei in grado di creare o utilizzare un server Chef, puoi visualizzare i messaggi di errore o i log per risolvere il problema. Le attività seguenti descrivono da dove iniziare per affrontare la risoluzione dei problemi relativi a un server Chef. Per informazioni su errori e soluzioni specifici, consulta la sezione Risoluzione di errori specifici di questo argomento.

- Usa la AWS OpsWorks for Chef Automate console per visualizzare i messaggi di errore se un server Chef non si avvia. Nella pagina dei dettagli del server Chef, i messaggi di errore relativi all'avvio e all'esecuzione del server vengono visualizzati nella parte superiore della pagina. Gli errori possono provenire da AWS OpsWorks for Chef Automate AWS CloudFormation, o da Amazon EC2, servizi utilizzati per creare un server Chef. Nella pagina dei dettagli puoi anche visualizzare gli eventi che si verificano su un server in esecuzione e che possono contenere messaggi relativi agli eventi di errore.

- Per risolvere i EC2 problemi, connettiti all'istanza del server utilizzando SSH e visualizza i log. EC2 i log delle istanze vengono archiviati nella directory. `/var/log/aws/opsworks-cm` Questi registri acquisiscono gli output dei comandi all' AWS OpsWorks for Chef Automate avvio di un server Chef.

Risoluzione di errori specifici

Argomenti

- Il server si trova in uno stato di interruzione della connessione
- Visualizzazione dei nodi gestiti nella colonna Missing (Mancante) del pannello di controllo di Chef Automate
- Impossibile creare un vault di Chef; il comando knife vault non viene eseguito e restituisce errori
- Creazione del server non riuscita e visualizzazione messaggio "requested configuration is currently not supported" (configurazione richiesta attualmente non supportata)
- Il server Chef non è in grado di riconoscere i nomi delle organizzazioni aggiunte nel pannello di controllo di Chef Automate
- Impossibile creare l' EC2 istanza Amazon del server
- Un errore del ruolo del servizio impedisce la creazione del server
- Superamento del limite di indirizzi IP elastici
- Impossibile effettuare l'accesso al pannello di controllo di Chef Automate
- Associazione automatica dei nodi non riuscita
- La manutenzione del sistema fallisce

Il server si trova in uno stato di interruzione della connessione

Problema: lo stato di un server viene visualizzato come Connessione interrotta.

Causa: ciò si verifica più comunemente quando un'entità esterna AWS OpsWorks apporta modifiche a un AWS OpsWorks for Chef Automate server o alle relative risorse di supporto. AWS OpsWorks non può connettersi ai server Chef Automate nello stato di connessione persa per gestire attività di manutenzione come la creazione di backup, l'applicazione di patch del sistema operativo o l'aggiornamento di Chef Automate. Di conseguenza, sul server potrebbero mancare aggiornamenti importanti, essere suscettibile di problemi di sicurezza o comunque non funzionare come previsto.

Soluzione: prova i seguenti passaggi per ripristinare la connessione del server.

1. Assicurati che il tuo ruolo di servizio disponga di tutte le autorizzazioni necessarie.

   a. Nella pagina Impostazioni del server, in Rete e sicurezza, scegli il link per il ruolo di servizio utilizzato dal server. Questo apre il ruolo di servizio per la visualizzazione nella console IAM.

   b. Nella scheda Autorizzazioni, verifica che `AWSOpsWorksCMServiceRole` sia presente nell'elenco delle politiche di autorizzazione. Se non è elencata, aggiungi manualmente la politica `AWSOpsWorksCMServiceRole` gestita al ruolo.

   c. Nella scheda Relazioni di fiducia, verifica che il ruolo di servizio disponga di una politica di fiducia che preveda che il `opsworks-cm.amazonaws.com` servizio assuma ruoli per tuo conto. Per ulteriori informazioni su come utilizzare le policy di fiducia con i ruoli, consulta Modifying a role (console) o il post del AWS Security Blog, How to use trust policies with IAM roles.

2. Assicurati che il profilo dell'istanza disponga di tutte le autorizzazioni necessarie.

   a. Nella pagina Impostazioni del server, in Rete e sicurezza, scegli il link per il profilo di istanza utilizzato dal server. Questo apre il profilo dell'istanza per la visualizzazione nella console IAM.

   b. Nella scheda Autorizzazioni, verifica che `AmazonEC2RoleforSSM` `AWSOpsWorksCMInstanceProfileRole` siano entrambe incluse nell'elenco delle politiche di autorizzazione. Se una o entrambe non sono elencate, aggiungi queste politiche gestite manualmente al ruolo.

   c. Nella scheda Relazioni di fiducia, verifica che il ruolo di servizio disponga di una politica di fiducia che preveda che il `ec2.amazonaws.com` servizio assuma ruoli per tuo conto. Per ulteriori informazioni su come utilizzare le policy di fiducia con i ruoli, consulta Modifying a role (console) o il post del AWS Security Blog, How to use trust policies with IAM roles.

3. Nella EC2 console Amazon, assicurati di trovarti nella stessa regione della regione del AWS OpsWorks for Chef Automate server, quindi riavvia l' EC2 istanza utilizzata dal server.

   a. Scegli l' EC2 istanza denominata `aws-opsworks-cm-instance-`*`server-name`*.

   b. Nel menu Stato dell'istanza, scegli Riavvia istanza.

   c. Attendi fino a 15 minuti prima che il server si riavvii e sia completamente online.

4. Nella AWS OpsWorks for Chef Automate console, nella pagina dei dettagli del server, verifica che lo stato del server sia ora integro.

Se lo stato del server è ancora Connessione interrotta dopo aver eseguito i passaggi precedenti, prova una delle seguenti soluzioni.

- Sostituisci il server creandone uno nuovo ed eliminando l'originale. Se i dati sul server corrente sono importanti per te, ripristina il server da un backup recente e verifica che i dati siano aggiornati prima di eliminare il server originale che non risponde.
- Contatta l'assistenza. AWS

Visualizzazione dei nodi gestiti nella colonna Missing (Mancante) del pannello di controllo di Chef Automate

Problema: un nodo gestito viene visualizzato nella colonna Missing (Mancante) del pannello di controllo di Chef Automate.

Causa: quando un nodo non si connette al server Chef Automate per più di 12 ore e `chef-client` non può essere eseguito sul nodo, il nodo assume lo stato che aveva nelle precedenti 12 ore e viene spostato nella colonna Missing (Mancante) del pannello di controllo di Chef Automate.

Soluzione: verifica che il nodo sia online. Prova a eseguire `knife node show` *`node_name`* `--run-list` per vedere se `chef-client` può essere eseguito sul nodo oppure `knife node show -l` *`node_name`* per visualizzare tutte le informazioni sul nodo. Il nodo potrebbe essere offline o disconnesso dalla rete.

Impossibile creare un vault di Chef; il comando **`knife vault`** non viene eseguito e restituisce errori

Problema: stai cercando di creare un vault sul server Chef Automate, ad esempio un vault per l'archiviazione delle credenziali per i nodi basati su Windows aggiunti al dominio, eseguendo il comando `knife vault`. Il comando restituisce un messaggio di errore simile al seguente.

```
WARN: Auto inflation of JSON data is deprecated. Please pass in the class to inflate or
 use #edit_hash (CHEF-1)
at /opt/chefdk/embedded/lib/ruby/2.3.0/forwardable.rb:189:in `edit_data'.Please see
 https://docs.chef.io/deprecations_json_auto_inflate.html
for further details and information on how to correct this problem.
WARNING: pivotal not found in users, trying clients.
ERROR: ChefVault::Exceptions::AdminNotFound: FATAL: Could not find pivotal in users or
 clients!
```

L'utente pivotal non viene restituito quando esegui `knife user list` in remoto, ma puoi visualizzare tale utente nei risultati quando esegui il comando `chef-server-ctl user-show` in locale sul server Chef Automate. In altre parole, il comando `knife vault` non è in grado di trovare l'utente pivotal, anche se sei sicuro della sua esistenza.

Causa: anche se l'utente pivotal viene considerato il superuser in Chef e pertanto dotato di autorizzazioni complete, tale utente non è membro di alcuna organizzazione, compresa l'organizzazione `default` utilizzata in AWS OpsWorks for Chef Automate. Il comando `knife user list` restituisce tutti gli utenti nell'organizzazione corrente nella configurazione di Chef. Il comando `chef-server-ctl user-show` restituisce tutti gli utenti, indipendentemente dall'organizzazione, compreso l'utente pivotal.

Soluzione: per risolvere il problema, aggiungi l'utente pivotal all'organizzazione predefinita eseguendo `knife opc`.

In primo luogo, dovrai installare il plugin knife-opc.

```
chef gem install knife-opc
```

Dopo aver installato il plugin, esegui il comando seguente per aggiungere l'utente pivotal all'organizzazione predefinita.

```
knife opc org user add default pivotal
```

Puoi verificare se l'utente pivotal fa parte dell'organizzazione predefinita eseguendo di nuovo `knife user list`. `pivotal` dovrebbe essere incluso nei risultati. A questo punto, esegui di nuovo `knife vault`.

Creazione del server non riuscita e visualizzazione messaggio "requested configuration is currently not supported" (configurazione richiesta attualmente non supportata)

Problema: stai cercando di creare un server Chef Automate, ma la creazione del server ha esito negativo e viene visualizzato un messaggio indicante che la configurazione richiesta attualmente non è supportata. Viene inoltre richiesto di controllare le configurazioni supportate nella documentazione.

Causa: è possibile che sia stato specificato un tipo di istanza non supportato per il server Chef Automate. Se scegli di creare il server Chef Automate in un VPC con una tenancy non predefinita, ad esempio una per le istanze dedicate, anche tutte le istanze all'interno del VPC specificato devono

avere un'istanza dedicata a tenant o host singolo. Poiché alcuni tipi di istanza, ad esempio t2, sono disponibili solo con la tenancy predefinita, il tipo di istanza del server Chef Automate potrebbe non essere supportato dal VPC specificato e pertanto la creazione del server ha esito negativo.

Soluzione: se scegli un VPC con una tenancy non predefinita, utilizza un tipo di istanza m4, che può supportare l'istanza dedicata a tenant singolo.

Il server Chef non è in grado di riconoscere i nomi delle organizzazioni aggiunte nel pannello di controllo di Chef Automate

Problema: hai aggiunti nuovi nomi di organizzazioni del flusso di lavoro nel pannello di controllo di Chef Automate oppure hai specificato un valore `CHEF_AUTOMATE_ORGANIZATION` diverso da `"default"` nello script di associazione automatica dei nodi, ma l'associazione dei nodi ha esito negativo. Il server AWS OpsWorks for Chef Automate non riconosce i nuovi nomi delle organizzazioni.

Causa: i nomi delle organizzazioni del flusso di lavoro e i nomi delle organizzazioni del server Chef non sono gli stessi. Puoi creare nuove organizzazioni del flusso di lavoro, ma non i nomi delle organizzazioni del server Chef, nel pannello di controllo basato sul Web di Chef Automate. Puoi utilizzare il pannello di controllo di Chef Automate solo per visualizzare le organizzazioni esistenti del server Chef. Una nuova organizzazione creata nel pannello di controllo di Chef Automate è un'organizzazione del flusso di lavoro e pertanto non viene riconosciuta dal server Chef. Non puoi creare nuovi nomi di organizzazione specificandoli nello script di associazione dei nodi. Il riferimento a un nome di organizzazione in uno script di associazione dei nodi causa la mancata esecuzione dell'associazione dei nodi se l'organizzazione non è stata prima aggiunta al server Chef.

Soluzione: per creare nuove organizzazioni che verranno riconosciute sul server Chef, utilizza il comando `knife opc org create` oppure esegui `chef-server-ctl org-create`.

Impossibile creare l' EC2 istanza Amazon del server

Problema: la creazione del server non è riuscita con un messaggio di errore simile al seguente: «Impossibile creare le seguenti risorse: [EC2Istanza]. Failed to receive 1 resource signal(s) within the specified duration." (Impossibile creare la seguente o le seguenti risorse: [EC2Instance]. Impossibile ricevere un segnale risorsa nell'intervallo specificato.)

Causa: Ciò è probabilmente dovuto al fatto che l' EC2istanza non ha accesso alla rete.

Soluzione: assicurati che l'istanza disponga dell'accesso a Internet in uscita e che l'agente AWS di servizio sia in grado di emettere comandi. Verifica che per il VPC (un VPC con una singola sottorete

pubblica) sia stata abilitata la funzionalità DNS resolution (Risoluzione DNS) e che per la sottorete sia stata abilitata l'impostazione Auto-assign Public IP (Assegna automaticamente IP pubblico).

Un errore del ruolo del servizio impedisce la creazione del server

Problema: la creazione del server non riesce e viene visualizzato un messaggio di errore che indica «Non autorizzato a eseguire sts:AssumeRole».

Causa: questa situazione si può verificare quando il ruolo del servizio in uso non dispone delle autorizzazioni adeguate per creare un nuovo server.

Soluzione: aprire la AWS OpsWorks for Chef Automate console e utilizzarla per generare un nuovo ruolo di servizio e un ruolo del profilo di istanza. Se preferisci utilizzare il tuo ruolo di servizio, allega la politica AWSOpsWorks CMService Role al ruolo. Verifica che opsworks-cm.amazonaws.com sia elencato tra i servizi nelle relazioni di fiducia del ruolo. Verifica che al ruolo di servizio associato al server Chef sia allegata la politica gestita di Works Role. AWSOps CMService

Superamento del limite di indirizzi IP elastici

Problema: la creazione del server non riesce e viene visualizzato un messaggio di errore che indica: «Impossibile creare le seguenti risorse: [EIP, EC2 Istanza]. Resource creation cancelled, the maximum number of addresses has been reached." (Impossibile creare la seguente o le seguenti risorse: [EIP, EC2Instance]. Creazione della risorsa annullata. È stato raggiunto il numero massimo di indirizzi)

Causa: questo potrebbe verificarsi quando l'account ha utilizzato il numero massimo di indirizzi IP elastici (EIP). Il limite predefinito di indirizzi EIP è cinque.

Soluzione: puoi rilasciare gli indirizzi EIP esistenti o eliminare quelli che il tuo account non utilizza attivamente oppure puoi contattare l'Assistenza AWS clienti per aumentare il limite di indirizzi EIP associato al tuo account.

Impossibile effettuare l'accesso al pannello di controllo di Chef Automate

Problema: la dashboard di Chef Automate mostra un errore simile al seguente: «Richiesta multiorigine bloccata: la stessa policy di origine non consente la lettura della risorsa remota inserita al momento https://myserver-name.region.opsworks-cm.io/api/v0/e/default/verify-token. (Reason: CORS header 'Access-Control-Allow-Origin' missing)". The error can also be similar to "The User Id / Password com dell'immissione non è corretta».

Causa: la dashboard di Chef Automate imposta esplicitamente il nome di dominio completo e non accetta il relativo. URLs Al momento, non puoi effettuare l'accesso utilizzando l'indirizzo IP del server Chef, ma a tale scopo puoi utilizzare il nome DNS del server.

Soluzione: accedi al pannello di controllo di Chef Automate solo utilizzando il nome DNS del server Chef, non il relativo indirizzo IP. Puoi anche provare a reimpostare le credenziali per il pannello di controllo di Chef Automate eseguendo un comando AWS CLI , come descritto in Reimpostazione delle credenziali del pannello di controllo di Chef Automate.

Associazione automatica dei nodi non riuscita

Problema: l'associazione automatica o incustodita di nuovi EC2 nodi Amazon non funziona. I nodi aggiunti al server Chef non vengono visualizzati nel pannello di controllo di Chef Automate e non vengono elencati nei risultati del comando `knife client show` o `knife node show`.

Causa: ciò può verificarsi quando non è impostato un ruolo IAM come profilo di istanza che consente alle chiamate `opsworks-cm` API di comunicare con nuove istanze. EC2

Soluzione: allega una policy al profilo dell' EC2 istanza che consenta di `DescribeNodeAssociationStatus` utilizzare `AssociateNode` le chiamate API EC2, come descritto in. Aggiungi nodi automaticamente in AWS OpsWorks for Chef Automate

La manutenzione del sistema fallisce

AWS OpsWorks CM esegue la manutenzione settimanale del sistema per garantire che le ultime versioni secondarie di Chef Server e Chef Automate Server, inclusi gli aggiornamenti di sicurezza, siano sempre in esecuzione su un server AWS OpsWorks for Chef Automate. Se, per qualsiasi motivo, la manutenzione del sistema fallisce, ti AWS OpsWorks CM avvisa dell'errore. Per ulteriori informazioni sulla manutenzione del sistema, vedereManutenzione del sistema in AWS OpsWorks for Chef Automate.

Questa sezione descrive i possibili motivi di errore e suggerisce soluzioni.

Argomenti

- Un errore nel profilo del ruolo o dell'istanza del servizio impedisce la manutenzione del sistema

Un errore nel profilo del ruolo o dell'istanza del servizio impedisce la manutenzione del sistema

Problema: la manutenzione del sistema non riesce con un messaggio di errore che indica «Non autorizzato a eseguire sts: AssumeRole «o un messaggio di errore simile relativo alle autorizzazioni.

Causa: ciò può verificarsi quando il ruolo di servizio o il profilo di istanza utilizzato non dispongono delle autorizzazioni adeguate per eseguire la manutenzione del sistema sul server.

Soluzione: assicurati che il ruolo di servizio e il profilo di istanza dispongano di tutte le autorizzazioni necessarie.

1. Assicurati che il tuo ruolo di servizio disponga di tutte le autorizzazioni necessarie.

    a. Nella pagina Impostazioni del server, in Rete e sicurezza, scegli il link per il ruolo di servizio utilizzato dal server. Questo apre il ruolo di servizio per la visualizzazione nella console IAM.

    b. Nella scheda Autorizzazioni, verifica che `AWSOpsWorksCMServiceRole` sia associato al ruolo di servizio. Se non `AWSOpsWorksCMServiceRole` è elencato, aggiungi questo criterio al ruolo.

    c. Verifica che opsworks-cm.amazonaws.com sia elencato tra i servizi nelle relazioni di fiducia del ruolo. Per ulteriori informazioni su come utilizzare le politiche di fiducia con i ruoli, consulta Modifying a role (console) o il post del AWS Security Blog, How to use trust policies with IAM roles.

2. Assicurati che il profilo dell'istanza disponga di tutte le autorizzazioni necessarie.

    a. Nella pagina Impostazioni del server, in Rete e sicurezza, scegli il link per il profilo di istanza utilizzato dal server. Questo apre il profilo dell'istanza per la visualizzazione nella console IAM.

    b. Nella scheda Autorizzazioni, verifica che `AmazonEC2RoleforSSM` `AWSOpsWorksCMInstanceProfileRole` siano entrambe incluse nell'elenco delle politiche di autorizzazione. Se una o entrambe non sono elencate, aggiungi queste politiche gestite manualmente al ruolo.

    c. Nella scheda Relazioni di fiducia, verifica che il ruolo di servizio disponga di una politica di fiducia che preveda che il `ec2.amazonaws.com` servizio assuma ruoli per tuo conto. Per ulteriori informazioni su come utilizzare le policy di fiducia con i ruoli, consulta Modifying a role (console) o il post del AWS Security Blog, How to use trust policies with IAM roles.

Supporto e informazioni della guida aggiuntivi

Se la risoluzione di un problema specifico non è descritta in questo argomento oppure se hai provato i suggerimenti riportati in questo argomento ma il problema persiste, visita i forum di AWS OpsWorks.

Puoi anche visitare il Centro AWS Support. Il AWS Support Center è l'hub per la creazione e la gestione dei casi di AWS assistenza. Il AWS Support Center include anche collegamenti ad altre risorse utili, come forum, informazioni tecniche FAQs, stato di salute del servizio e AWS Trusted Advisor.

Sicurezza nella gestione della AWS OpsWorks configurazione (CM)

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. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei programmi di conformitàAWS. Per ulteriori informazioni sui programmi di conformità che si applicano ad AWS OpsWorks CM, consulta Servizi coperti dal programma di conformitàAWS.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizza AWS OpsWorks CM. I seguenti argomenti mostrano come configurare AWS OpsWorks CM per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri servizi AWS che ti aiutano a monitorare e proteggere le tue risorse AWS OpsWorks CM.

Argomenti

- Protezione dei dati in AWS OpsWorks CM
- Crittografia dei dati
- Identity and Access Management per AWS OpsWorks CM
- Riservatezza del traffico Internet
- Registrazione e monitoraggio in CM AWS OpsWorks
- Convalida della conformità per AWS OpsWorks CM
- Resilienza nel CM AWS OpsWorks
- Sicurezza dell'infrastruttura in AWS OpsWorks CM
- Analisi della configurazione e delle vulnerabilità in CM AWS OpsWorks

- Best practice di sicurezza per CM AWS OpsWorks

Protezione dei dati in AWS OpsWorks CM

Il modello di responsabilità AWS condivisa modello di si applica alla protezione dei dati in AWS OpsWorks Configuration Management. 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, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS 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 con OpsWorks CM o altri utenti Servizi AWS utilizzando la console, l'API o. AWS CLI AWS SDKs 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 fornisci un URL a un

server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

I nomi dei server OpsWorks CM non sono crittografati.

OpsWorks CM raccoglie i seguenti dati sui clienti durante la creazione e la manutenzione dei server e dei AWS OpsWorks for Puppet Enterprise server. AWS OpsWorks for Chef Automate

- OpsWorks Per Puppet Enterprise, raccogliamo le chiavi private che Puppet Enterprise utilizza per abilitare la comunicazione tra il tuo Puppet master e i nodi gestiti.
- Ad AWS OpsWorks for Chef Automate esempio, raccogliamo chiavi private per i certificati che alleghi al servizio se utilizzi un dominio personalizzato. La chiave privata fornita durante la creazione di un server Chef Automate con un dominio personalizzato viene passata al server.

OpsWorks I server CM memorizzano il codice di configurazione, ad esempio i libri di cucina Chef o i moduli Puppet Enterprise. Sebbene questo codice sia memorizzato nei backup dei server, AWS non vi ha accesso. Questo contenuto è crittografato e solo gli amministratori del tuo AWS account possono accedervi. Ti consigliamo di proteggere il codice di configurazione Chef o Puppet utilizzando i protocolli consigliati per i repository di origine. Ad esempio, è possibile limitare le autorizzazioni agli archivi di o seguire le linee guida sul GitHub sito Web per proteggere gli archivi. AWS CodeCommit GitHub

OpsWorks CM non utilizza i contenuti forniti dai clienti per mantenere il servizio o conservare i registri dei clienti. I log relativi ai tuoi server OpsWorks CM vengono archiviati nel tuo account, nei bucket Amazon S3. Gli indirizzi IP degli utenti che si connettono ai tuoi server OpsWorks CM vengono registrati da. AWS

Integrazione con AWS Secrets Manager

A partire dal 3 maggio 2021, quando crei un nuovo server in OpsWorks CM, OpsWorks CM archivia i segreti del server in AWS Secrets Manager. Per i nuovi server, i seguenti attributi vengono archiviati come segreti in Secrets Manager.

- Server Chef Automate
  - Chiave privata HTTPS (solo server che non utilizzano un dominio personalizzato)
  - Password amministrativa di Chef Automate (CHEF_AUTOMATE_ADMIN_PASSWORD)
- Maestro di Puppet Enterprise
  - Chiave privata HTTPS (solo server che non utilizzano un dominio personalizzato)

- Password amministrativa Puppet (PUPPET_ADMIN_PASSWORD)
- Telecomando Puppet r10k (PUPPET_R10K_REMOTE)

Per i server esistenti che non utilizzano un dominio personalizzato, l'unico segreto archiviato in Secrets Manager, sia per i server Chef Automate che per quelli Puppet Enterprise, è la chiave privata HTTPS, poiché viene generata durante la manutenzione automatica settimanale del sistema.

OpsWorks CM archivia automaticamente i segreti in Secrets Manager e questo comportamento non è configurabile dall'utente.

Crittografia dei dati

AWS OpsWorks CM crittografa i backup dei server e le comunicazioni tra AWS gli utenti autorizzati e i relativi server CM. AWS OpsWorks Tuttavia, i volumi root Amazon EBS dei server AWS OpsWorks CM non sono crittografati.

Crittografia dei dati inattivi

AWS OpsWorks I backup dei server CM sono crittografati. Tuttavia, i volumi root Amazon EBS dei server AWS OpsWorks CM non sono crittografati. Questa opzione non è configurabile dall'utente.

Crittografia in transito

AWS OpsWorks CM utilizza HTTP con crittografia TLS. AWS OpsWorks CM utilizza per impostazione predefinita i certificati autofirmati per il provisioning e la gestione dei server, se gli utenti non forniscono alcun certificato firmato. Si consiglia di utilizzare un certificato firmato di un'autorità di certificazione (CA).

Gestione delle chiavi

AWS Key Management Service le chiavi gestite dai clienti e le chiavi gestite da AWS non sono attualmente supportate da AWS OpsWorks CM.

Identity and Access Management per AWS OpsWorks CM

AWS Identity and Access Management (IAM) è un AWS servizio che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può

essere autenticato (effettuato l'accesso) e autorizzato (dispone delle autorizzazioni) a utilizzare le risorse CM. OpsWorks IAM è un AWS servizio che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso tramite policy
- Come funziona AWS OpsWorks CM con IAM
- AWS OpsWorks Esempi di policy basate sull'identità CM
- Risoluzione dei problemi relativi a AWS OpsWorks CM Identity and Access
- AWS politiche gestite per la gestione della AWS OpsWorks configurazione
- Prevenzione intersettoriale confusa dei sostituti in AWS OpsWorks CM

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in OpsWorks CM.

Utente del servizio: se utilizzi il servizio OpsWorks CM per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più funzionalità OpsWorks CM per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in OpsWorks CM, consultaRisoluzione dei problemi relativi a AWS OpsWorks CM Identity and Access.

Amministratore del servizio: se sei responsabile delle risorse OpsWorks CM della tua azienda, probabilmente hai pieno accesso a OpsWorks CM. È tuo compito determinare a quali funzionalità e risorse OpsWorks CM devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con OpsWorks CM, consultaCome funziona AWS OpsWorks CM con IAM.

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere policy per gestire l'accesso a OpsWorks CM. Per visualizzare esempi di policy basate

sull'identità OpsWorks CM che puoi utilizzare in IAM, consulta. AWS OpsWorks Esempi di policy basate sull'identità CM

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sul metodo consigliato per la firma delle richieste, consulta Signature Version 4 AWS per le richieste API nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA)AWS in IAM nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root

può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente IAM.

> **Warning**
>
> Gli utenti IAM dispongono di credenziali a lungo termine, il che rappresenta un rischio per la sicurezza. Per ridurre questo rischio, si consiglia di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Per assumere temporaneamente un ruolo IAM in AWS Management Console, puoi passare da un ruolo utente a un ruolo IAM (console). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o

utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Create a role for a third-party identity provider (federation) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze 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 IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso inoltrato (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un' EC2 istanza e che AWS CLI effettuano richieste AWS API. Questa soluzione è preferibile alla memorizzazione delle chiavi di accesso all'interno dell' EC2 istanza. Per assegnare un AWS ruolo a un' EC2 istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull' EC2 istanza di ottenere credenziali temporanee. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella IAM User Guide.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate sulle identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

OpsWorks CM supporta policy personalizzate create in IAM e associate a utenti, ruoli o gruppi.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

OpsWorks CM non supporta politiche basate sulle risorse.

Elenchi di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la panoramica della lista di controllo degli accessi (ACL) nella Amazon Simple Storage Service Developer Guide.

OpsWorks CM non utilizza ACLs.

Altri tipi di policy

OpsWorks CM non supporta i seguenti altri tipi di policy.

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite di autorizzazioni è una funzionalità avanzata in cui si impostano le autorizzazioni massime che una policy basata sull'identità può concedere a un'entità IAM (utente o ruolo). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i suoi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.
- Politiche di controllo del servizio (SCPs): SCPs sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più AWS account di proprietà dell'azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità presenti negli account dei membri, inclusa ciascuna di esse. Utente root dell'account AWS Per ulteriori informazioni su Organizations and SCPs, consulta How SCPs Work nella AWS Organizations User Guide.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire

da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta la logica di valutazione delle policy nella IAM User Guide.

Come funziona AWS OpsWorks CM con IAM

Prima di utilizzare IAM per gestire l'accesso a AWS OpsWorks CM, è necessario comprendere quali funzionalità IAM sono disponibili per l'uso con AWS OpsWorks CM. Per avere una visione di alto livello di come AWS OpsWorks CM e altri AWS servizi funzionano con IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Argomenti

- AWS OpsWorks Politiche basate sull'identità di CM
- AWS OpsWorks CM e politiche basate sulle risorse
- Autorizzazione basata sui tag CM AWS OpsWorks
- AWS OpsWorks Ruoli CM IAM

AWS OpsWorks Politiche basate sull'identità di CM

Con le policy basate sull'identità IAM, puoi specificare azioni e risorse consentite o negate, nonché le condizioni in base alle quali le azioni sono consentite o negate. AWS OpsWorks CM supporta azioni, risorse e chiavi di condizione specifiche. Per informazioni su tutti gli elementi utilizzati in una policy JSON, consulta Documentazione di riferimento degli elementi delle policy JSON IAM nella Guida per l'utente IAM.

In AWS OpsWorks CM, è possibile allegare una dichiarazione politica personalizzata a un utente, ruolo o gruppo.

Operazioni

L'elemento `Action` di una policy basata su identità IAM descrive l'operazione o le operazioni specifiche che saranno concesse o rifiutate dalla policy. Le azioni politiche in genere hanno lo stesso

nome dell'operazione AWS API associata. L'operazione viene utilizzata in una policy per concedere le autorizzazioni di eseguire l'operazione associata.

Le azioni politiche in AWS OpsWorks CM utilizzano il seguente prefisso prima dell'azione:`opsworks-cm:`. Ad esempio, per concedere a qualcuno l'autorizzazione per creare un server AWS OpsWorks CM mediante l'operazione API, includi l'operazione `opsworks-cm:CreateServer` nella policy. Le dichiarazioni politiche devono includere un `NotAction` elemento `Action` or. AWS OpsWorks CM 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": [
      "opsworks-cm:action1",
      "opsworks-cm:action2"
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Describe`, includi la seguente azione:

```
"Action": "opsworks-cm:Describe*"
```

Quando utilizzi i caratteri jolly per consentire più operazioni in un'istruzione di policy, presta attenzione a consentire le operazioni solo per i servizi o gli utenti autorizzati.

Per visualizzare un elenco di azioni AWS OpsWorks CM, consulta Actions, Resources and Condition Keys for AWS OpsWorks nella IAM User Guide.

Risorse

L'elemento `Resource` specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Specifica una risorsa utilizzando un ARN o il carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

Puoi ottenere l'Amazon Resource Number (ARN) di un server AWS OpsWorks CM o di un backup eseguendo le operazioni DescribeServerso le DescribeBackupsAPI e basare politiche a livello di risorsa su tali risorse.

Una risorsa del server AWS OpsWorks CM ha un ARN nel seguente formato:

```
arn:aws:opsworks-cm:{Region}:${Account}:server/${ServerName}/${UniqueId}
```

Una risorsa di backup AWS OpsWorks CM ha un ARN nel seguente formato:

```
arn:aws:opsworks-cm:{Region}:${Account}:backup/${ServerName}-{Date-and-Time-Stamp-of-Backup}
```

Per ulteriori informazioni sul formato di ARNs, consulta Amazon Resource Names (ARNs) e AWS Service Namespaces.

Ad esempio, per specificare il server `test-chef-automate` Chef Automate nell'istruzione, utilizza il seguente ARN:

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:server/test-chef-automate/EXAMPLE-d1a2bEXAMPLE"
```

Per specificare tutti i server AWS OpsWorks CM che appartengono a un account specifico, usa il carattere jolly (*):

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:server/*"
```

L'esempio seguente specifica un backup del server AWS OpsWorks CM come risorsa:

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:backup/test-chef-automate-server-2018-05-20T19:06:12.399Z"
```

Alcune azioni AWS OpsWorks CM, come quelle per la creazione di risorse, non possono essere eseguite su una risorsa specifica. In questi casi, è necessario utilizzare il carattere jolly (*).

```
"Resource": "*"
```

Molte operazioni API di coinvolgono più risorse. Per specificare più risorse in un'unica istruzione, separale ARNs con virgole.

```
"Resource": [
      "resource1",
      "resource2"
```

Per visualizzare un elenco dei tipi di risorse AWS OpsWorks CM e relativi ARNs, consulta Actions, Resources and Condition Keys for AWS OpsWorks CM nella IAM User Guide. Per sapere con quali azioni è possibile specificare l'ARN di ogni risorsa, consulta Actions, Resources, and Condition Keys for AWS OpsWorks CM nella IAM User Guide.

Chiavi di condizione

AWS OpsWorks CM non dispone di chiavi contestuali specifiche del servizio che possono essere utilizzate nell'`Condition`elemento delle dichiarazioni politiche. Per l'elenco delle chiavi di contesto globali disponibili per tutti i servizi, consulta le chiavi di contesto delle condizioni AWS globali nello IAM Policy Reference. Per vedere tutte le chiavi di contesto delle condizioni AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. Puoi compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se specifichi più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'`0R`operazione logica. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile concedere a un utente l'autorizzazione ad accedere a una risorsa solo se è contrassegnata con il nome dell'utente. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

Esempi

Per visualizzare esempi di politiche basate sull'identità AWS OpsWorks CM, vedere. AWS OpsWorks Esempi di policy basate sull'identità CM

AWS OpsWorks CM e politiche basate sulle risorse

AWS OpsWorks CM non supporta politiche basate sulle risorse.

Le policy basate su risorse sono documenti di policy JSON che specificano le operazioni che possono essere eseguite da un'entità principale specificata sulla risorsa e in base a quali condizioni.

Autorizzazione basata sui tag CM AWS OpsWorks

È possibile allegare tag alle risorse AWS OpsWorks CM o passare i tag in una richiesta a AWS OpsWorks CM. Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:RequestTag/key-name` o `aws:TagKeys`. Per ulteriori informazioni sull'etichettatura delle risorse AWS OpsWorks CM, consulta Lavorare con i tag sulle AWS OpsWorks for Chef Automate risorse o Lavorare con i tag sulle AWS OpsWorks for Puppet Enterprise risorse consulta questa guida.

AWS OpsWorks Ruoli CM IAM

Un ruolo IAM è un'entità all'interno del tuo AWS account che dispone di autorizzazioni specifiche.

AWS OpsWorks CM utilizza due ruoli:

- Un ruolo di servizio che concede al servizio AWS OpsWorks CM le autorizzazioni per funzionare all'interno dell'account di AWS un utente. Se si utilizza il ruolo di servizio predefinito fornito da OpsWorks CM, il nome di questo ruolo è. `aws-opsworks-cm-service-role`
- Un ruolo di profilo di istanza che consente al servizio AWS OpsWorks CM di chiamare l'API OpsWorks CM. Questo ruolo consente l'accesso ad Amazon S3 AWS CloudFormation e la creazione del server e del bucket S3 per i backup. Se utilizzi il profilo di istanza predefinito fornito da OpsWorks CM, il nome di questo ruolo del profilo di istanza è. `aws-opsworks-cm-ec2-role`

AWS OpsWorks CM non utilizza ruoli collegati ai servizi.

Utilizzo di credenziali temporanee con AWS OpsWorks CM

AWS OpsWorks CM supporta l'utilizzo di credenziali temporanee e eredita tale funzionalità da. AWS Security Token Service

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 o. AssumeRoleGetFederationToken

Ruoli collegati al servizio

AWS OpsWorks CM non utilizza ruoli collegati ai servizi.

I ruoli collegati ai AWS servizi consentono ai servizi di accedere alle risorse di altri servizi per completare un'azione per conto dell'utente. I ruoli collegati ai servizi sono visualizzati nell'account IAM

e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non può modificarle.

Ruoli dei servizi

Questa caratteristica 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 dei servizi sono visualizzati nell'account IAM e sono di proprietà dell'account. Ciò significa che un amministratore IAM può modificare le autorizzazioni per questo ruolo. Tuttavia, questo potrebbe pregiudicare la funzionalità del servizio.

AWS OpsWorks CM utilizza due ruoli:

- Un ruolo di servizio che concede al servizio AWS OpsWorks CM le autorizzazioni per funzionare all'interno dell'account di AWS un utente. Se si utilizza il ruolo di servizio predefinito fornito da OpsWorks CM, il nome di questo ruolo è. `aws-opsworks-cm-service-role`
- Un ruolo di profilo di istanza che consente al servizio AWS OpsWorks CM di chiamare l'API OpsWorks CM. Questo ruolo consente l'accesso ad Amazon S3 AWS CloudFormation e la creazione del server e del bucket S3 per i backup. Se utilizzi il profilo di istanza predefinito fornito da OpsWorks CM, il nome di questo ruolo del profilo di istanza è. `aws-opsworks-cm-ec2-role`

Scelta di un ruolo IAM in AWS OpsWorks CM

Quando crei un server in AWS OpsWorks CM, devi scegliere un ruolo per consentire a AWS OpsWorks CM di accedere ad Amazon per tuo EC2 conto. Se hai già creato un ruolo di servizio, AWS OpsWorks CM ti fornisce un elenco di ruoli tra cui scegliere. OpsWorks CM può creare il ruolo per te, se non ne specifichi uno. È importante scegliere un ruolo che consenta l'accesso per avviare e arrestare EC2 le istanze Amazon. Per ulteriori informazioni, consulta Creare un server Chef Automate o Creare un Puppet Enterprise Master.

AWS OpsWorks Esempi di policy basate sull'identità CM

Per impostazione predefinita, gli utenti o i ruoli non dispongono dell'autorizzazione per creare o modificare AWS OpsWorks risorse CM. Inoltre, non possono eseguire attività utilizzando l' AWS API AWS Management Console AWS CLI, o. Un amministratore IAM deve creare policy IAM che concedano alle identità IAM il permesso di eseguire operazioni API specifiche sulle risorse specifiche di cui ha bisogno. L'amministratore devi quindi collegare queste policy 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, consulta Creazione di policy nella scheda JSON nella Guida per l'utente di IAM.

In AWS OpsWorks CM, puoi assegnare la `AWSOpsWorksCMServiceRole` policy a un utente per consentirgli di creare e gestire i server Chef Automate o Puppet Enterprise utilizzando o. AWS Management Console AWS CLI

Argomenti

- Best practice delle policy
- Consenti agli utenti di visualizzare le loro autorizzazioni
- Visualizzazione dei server AWS OpsWorks CM in base ai tag

Best practice delle policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare OpsWorks risorse CM nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui 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 gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS 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 ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi 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 privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue 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 AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.

- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente 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, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Protezione dell'accesso API con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Consenti agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa policy include le autorizzazioni per completare questa azione sulla console o utilizzando programmaticamente l'API o. 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": "*"
        }
    ]
}
```

Visualizzazione dei server AWS OpsWorks CM in base ai tag

È possibile utilizzare le condizioni della politica basata sull'identità per controllare l'accesso ai server AWS OpsWorks CM e ai backup basati sui tag. Questo esempio mostra come è possibile creare una politica che consenta la visualizzazione di un server CM. AWS OpsWorks Tuttavia, l'autorizzazione viene concessa solo se il tag del server AWS OpsWorks CM `Owner` ha il valore del nome utente di quell'utente. Questa policy concede anche le autorizzazioni necessarie per completare questa azione nella console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListServersInConsole",
            "Effect": "Allow",
            "Action": "opsworks-cm:DescribeServers",
            "Resource": "*"
        },
        {
            "Sid": "ViewServerIfOwner",
            "Effect": "Allow",
            "Action": "opsworks-cm:DescribeServers",
            "Resource": "arn:aws:opsworks-cm:region:master-account-ID:server/server-
name",
            "Condition": {
                "StringEquals": {"opsworks-cm:ResourceTag/Owner": "${aws:username}"}
```

```
            }
        }
    ]
}
```

È possibile collegare questa policy agli utenti nell'account. Se un utente denominato `richard-roe` tenta di visualizzare un server AWS OpsWorks CM, al server deve essere assegnato il tag `0wner=richard-roe` o`owner=richard-roe`. In caso contrario l'accesso è negato. La chiave di tag di condizione `0wner` corrisponde a `0wner` e `owner` perché i nomi delle chiavi di condizione non effettuano la distinzione tra maiuscole e minuscole. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.

Risoluzione dei problemi relativi a AWS OpsWorks CM Identity and Access

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con IAM. Per informazioni sulla risoluzione dei problemi specifiche di AWS OpsWorks CM, consulta Risoluzione dei problemi AWS OpsWorks for Chef Automate eRisoluzione dei problemi OpsWorks per Puppet Enterprise.

Argomenti

- Non sono autorizzato a eseguire un'azione in AWS OpsWorks CM
- Non sono autorizzato a eseguire iam: PassRole
- Voglio consentire a persone esterne al mio AWS account di accedere alle mie risorse AWS OpsWorks CM

Non sono autorizzato a eseguire un'azione in AWS OpsWorks CM

Se ti AWS Management Console dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è colui che ti ha fornito le credenziali di accesso.

L'errore di esempio seguente si verifica quando l'utente `mateojackson` tenta di utilizzare la console per visualizzare i dettagli su un server AWS OpsWorks CM, ma non dispone `opsworks-cm:DescribeServers` delle autorizzazioni.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 opsworks-cm:DescribeServers on resource: test-chef-automate-server
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa `test-chef-automate-server` mediante l'operazione `opsworks-cm:DescribeServers`.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, devi contattare il tuo amministratore per ricevere assistenza. L'amministratore è colui che ti ha fornito le credenziali di accesso. Chiedi a quella persona di aggiornare le tue politiche per consentirti di trasferire un ruolo a OpsWorks CM.

Alcuni AWS servizi consentono di trasferire un ruolo esistente a quel servizio, anziché 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 denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in OpsWorks CM. Tuttavia, l'operazione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone di autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, Mary chiede all'amministratore di aggiornare la sue policy per poter eseguire l'operazione `iam:PassRole`.

Voglio consentire a persone esterne al mio AWS account di accedere alle mie risorse AWS OpsWorks CM

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 i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- AWS OpsWorks CM supporta la concessione agli utenti di più di un account di accedere alla gestione di un server CM. AWS OpsWorks

- Per scoprire come fornire l'accesso alle tue risorse su tutti AWS gli account di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro AWS account di tua proprietà nella Guida per l'utente IAM.
- Per scoprire come fornire l'accesso alle tue risorse ad AWS account di terze parti, consulta Fornire l'accesso agli AWS account di proprietà di terze parti nella IAM User Guide.
- 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 tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consultare Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

AWS politiche gestite per la gestione della AWS OpsWorks configurazione

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più facile utilizzare le politiche AWS gestite che scrivere le politiche autonomamente. Creare policy gestite dal cliente IAM per fornire al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste politiche coprono casi d'uso comuni e sono disponibili nel tuo AWS account. Per ulteriori informazioni sulle policy AWS gestite, consulta le policy AWS gestite nella IAM User Guide.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi occasionalmente aggiungono altre autorizzazioni a una policy gestita da AWS per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una policy gestita da AWS quando viene avviata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non comprometteranno le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy ReadOnlyAccess AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consulta la sezione Policy gestite da AWS per funzioni di processi nella Guida per l'utente di IAM.

Policy gestita da AWS: **AWSOpsWorksCMServiceRole**

Puoi collegarti `AWSOpsWorksCMServiceRole` alle tue entità IAM. OpsWorks CM attribuisce questa politica anche a un ruolo di servizio che consente a OpsWorks CM di eseguire azioni per conto dell'utente.

Questa politica concede *`administrative`* autorizzazioni che consentono agli amministratori OpsWorks CM di creare, gestire ed eliminare server e backup OpsWorks CM.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `opsworks-cm`— Consente ai responsabili di eliminare i server esistenti e avviare le operazioni di manutenzione.
- `acm`— Consente ai responsabili di eliminare o importare certificati AWS Certificate Manager che consentono agli utenti di connettersi a un server OpsWorks CM.
- `cloudformation`— Consente a OpsWorks CM di creare e gestire gli AWS CloudFormation stack quando i principali creano, aggiornano o OpsWorks eliminano i server CM.
- `ec2`— Consente a OpsWorks CM di avviare, fornire, aggiornare e terminare le istanze di Amazon Elastic Compute Cloud quando i responsabili creano, aggiornano o eliminano server CM. OpsWorks
- `iam`— Consente a OpsWorks CM di creare ruoli di servizio necessari per la creazione e la gestione dei server CM. OpsWorks
- `tag`— Consente ai responsabili di applicare e rimuovere tag dalle risorse OpsWorks CM, inclusi server e backup.
- `s3`— Consente a OpsWorks CM di creare bucket Amazon S3 per archiviare i backup dei server, gestire oggetti nei bucket S3 su richiesta principale (ad esempio, l'eliminazione di un backup) ed eliminare i bucket.
- `secretsmanager`— Consente a OpsWorks CM di creare e gestire i segreti di Secrets Manager e di applicare o rimuovere tag dai segreti.
- `ssm`— Consente a OpsWorks CM di utilizzare Systems Manager Run Command sulle istanze che sono server OpsWorks CM.

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

```
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket*"
            ],
            "Action": [
                "s3:CreateBucket",
                "s3:DeleteObject",
                "s3:DeleteBucket",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutBucketPolicy",
                "s3:PutObject",
                "s3:GetBucketTagging",
                "s3:PutBucketTagging"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Action": [
                "tag:UntagResources",
                "tag:TagResources"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Action": [
                "ssm:DescribeInstanceInformation",
                "ssm:GetCommandInvocation",
                "ssm:ListCommandInvocations",
                "ssm:ListCommands"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
```

```
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:cloudformation:stack-name": "aws-opsworks-cm-
*"
                }
            },
            "Action": [
                "ssm:SendCommand"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*::document/*",
                "arn:aws:s3:::amzn-s3-demo-bucket*"
            ],
            "Action": [
                "ssm:SendCommand"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Action": [
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateImage",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSnapshot",
                "ec2:CreateTags",
                "ec2:DeleteSecurityGroup",
                "ec2:DeleteSnapshot",
                "ec2:DeregisterImage",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstances",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSnapshots",
                "ec2:DescribeSubnets",
```

```
                "ec2:DisassociateAddress",
                "ec2:ReleaseAddress",
                "ec2:RunInstances",
                "ec2:StopInstances"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "ec2:ResourceTag/aws:cloudformation:stack-name": "aws-opsworks-cm-
*"
                }
            },
            "Action": [
                "ec2:TerminateInstances",
                "ec2:RebootInstances"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:opsworks-cm:*:*:server/*"
            ],
            "Action": [
                "opsworks-cm:DeleteServer",
                "opsworks-cm:StartMaintenance"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/aws-opsworks-cm-*"
            ],
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:UpdateStack"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::*:role/aws-opsworks-cm-*",
                "arn:aws:iam::*:role/service-role/aws-opsworks-cm-*"
            ],
            "Action": [
                "iam:PassRole"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "acm:DeleteCertificate",
                "acm:ImportCertificate"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "arn:aws:secretsmanager:*:*:opsworks-cm!aws-opsworks-cm-
secrets-*",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:UpdateSecret",
                "secretsmanager:DeleteSecret",
                "secretsmanager:TagResource",
                "secretsmanager:UntagResource"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteTags",
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:elastic-ip/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        }
    ]
```

```
}
```

Policy gestita da AWS: **AWSOpsWorksCMInstanceProfileRole**

Puoi collegarti `AWSOpsWorksCMInstanceProfileRole` alle tue entità IAM. OpsWorks CM attribuisce questa politica anche a un ruolo di servizio che consente a OpsWorks CM di eseguire azioni per conto dell'utente.

Questa politica concede *`administrative`* autorizzazioni che consentono alle EC2 istanze Amazon utilizzate come server OpsWorks CM di ottenere informazioni da AWS CloudFormation e AWS Secrets Manager archiviare i backup dei server nei bucket Amazon S3.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `acm`— Consente alle EC2 istanze del server OpsWorks CM di ottenere certificati AWS Certificate Manager che consentono agli utenti di connettersi a un server CM. OpsWorks
- `cloudformation`— Consente alle EC2 istanze del server OpsWorks CM di ottenere informazioni sugli AWS CloudFormation stack durante il processo di creazione o aggiornamento dell'istanza e di inviare segnali AWS CloudFormation sul relativo stato.
- `s3`— Consente alle EC2 istanze del server OpsWorks CM di caricare e archiviare i backup dei server nei bucket S3, interrompere o ripristinare i caricamenti se necessario ed eliminare i backup dai bucket S3.
- `secretsmanager`— Consente alle EC2 istanze del server OpsWorks CM di ottenere i valori dei segreti di Secrets Manager relativi a OpsWorks CM.

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

```
            ]
        },
        {
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:ListMultipartUploadParts",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket*",
            "Effect": "Allow"
        },
        {
            "Action": "acm:GetCertificate",
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "arn:aws:secretsmanager:*:*:opsworks-cm!aws-opsworks-cm-
secrets-*",
            "Effect": "Allow"
        }
    ]
}
```

OpsWorks Aggiornamenti CM alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per OpsWorks CM da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei documenti OpsWorks CM.

Modifica	Descrizione	Data
AWSOpsWorks CMInstance ProfileRole - Politica gestita aggiornata	OpsWorks CM ha aggiornato la policy gestita che consente alle EC2 istanze utilizzat e come server OpsWorks	23 Aprile 2021

Modifica	Descrizione	Data
	CM di condividere informazioni con CloudFormation Secrets Manager e gestire i backup. La modifica aggiunge `opsworks-cm!` il nome della risorsa per i segreti di Secrets Manager, in modo che OpsWorks CM possa possedere i segreti.	
AWSOpsWorks CMService Role: politica gestita aggiornata	OpsWorks CM ha aggiornato la politica gestita che consente agli amministratori OpsWorks CM di creare, gestire ed eliminare server e backup OpsWorks CM. La modifica aggiunge `opsworks-cm!` il nome della risorsa per i segreti di Secrets Manager, in modo che OpsWorks CM possa possedere i segreti.	23 Aprile 2021
OpsWorks CM ha iniziato a tenere traccia delle modifiche	OpsWorks CM ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	23 Aprile 2021

Prevenzione intersettoriale confusa dei sostituti in AWS OpsWorks CM

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Nel AWS, l'impersonificazione tra servizi può portare alla confusione del vicesceriffo. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a 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 ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account.

Si consiglia di utilizzare `aws:SourceArn`le chiavi di contesto della condizione `aws:SourceAccount`globale nelle politiche delle risorse per limitare le autorizzazioni che AWS OpsWorks CM forniscono un altro servizio alla risorsa. Se il valore `aws:SourceArn` non contiene l'ID account, ad esempio un ARN di un bucket Amazon S3, è necessario utilizzare entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni. 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 consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

Il valore di `aws:SourceArn` deve essere l'ARN di un server OpsWorks CM Chef o Puppet.

Il modo più efficace per proteggersi dal confuso problema vice consiste nell'utilizzare la chiave di contesto della condizione `aws:SourceArn` globale con l'ARN completo del AWS OpsWorks CM server. Se non conosci l'ARN completo o se stai specificando più server ARNs, usa la chiave `aws:SourceArn` global context condition con wildcards (*) per le parti sconosciute dell'ARN. Ad esempio `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

La sezione seguente mostra come utilizzare le chiavi di contesto della `aws:SourceArn` condizione `aws:SourceAccount` globale AWS OpsWorks CM per evitare il confuso problema del vice.

Evita gli exploit confusi dei vice in AWS OpsWorks CM

Questa sezione descrive come contribuire a prevenire la confusione degli exploit dei deputati AWS OpsWorks CM e include esempi di policy di autorizzazione che puoi associare al ruolo IAM a cui stai accedendo. AWS OpsWorks CM Come best practice in materia di sicurezza, ti consigliamo di aggiungere le chiavi `aws:SourceArn` and `aws:SourceAccount` condition alle relazioni di fiducia che il tuo ruolo IAM intrattiene con altri servizi. Le relazioni di fiducia AWS OpsWorks CM consentono di assumere il ruolo di eseguire azioni in altri servizi necessari per creare o gestire i AWS OpsWorks CM server.

Modificare le relazioni di fiducia per aggiungere **`aws:SourceArn`** e **`aws:SourceAccount`** condizionare le chiavi

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione a sinistra, seleziona Ruoli.
3. Nella casella di ricerca, cerca il ruolo a cui accedi AWS OpsWorks CM. Il ruolo AWS gestito è`aws-opsworks-cm-service-role`.
4. Nella pagina di riepilogo del ruolo, scegli la scheda Relazioni di fiducia.
5. Nella scheda Relazioni di attendibilità, scegli Modifica relazione di attendibilità.
6. Nel documento Policy, aggiungi almeno una delle chiavi `aws:SourceArn` o delle `aws:SourceAccount` condizioni alla politica. `aws:SourceArn`Utilizzalo per limitare la relazione di fiducia tra servizi diversi (ad esempio AWS Certificate Manager e Amazon EC2) e AWS OpsWorks CM a AWS OpsWorks CM server specifici, il che è più restrittivo. Aggiungi `aws:SourceAccount` per limitare la relazione di fiducia tra servizi diversi e AWS OpsWorks CM ai server di un account specifico, in modo meno restrittivo. Di seguito è riportato un esempio. Tieni presente che se utilizzi entrambe le chiavi di condizione, l'account IDs deve essere lo stesso.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "opsworks-cm.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:opsworks-cm:us-east-2:123456789012:server/my-
opsworks-server/EXAMPLEabcd-1234-efghEXAMPLE-ID"
        }
      }
    }
  ]
}
```

7. Quando hai finito di aggiungere le chiavi di condizione, scegli Aggiorna politica di attendibilità.

Di seguito sono riportati altri esempi di ruoli che limitano l'accesso ai AWS OpsWorks CM server utilizzando `aws:SourceArn` and`aws:SourceAccount`.

Argomenti

- Esempio: accesso ai AWS OpsWorks CM server in una regione specifica
- Esempio: aggiunta di più di un ARN del server a aws:SourceArn

Esempio: accesso ai AWS OpsWorks CM server in una regione specifica

La seguente dichiarazione sulla relazione di fiducia tra ruoli accede a qualsiasi AWS OpsWorks CM server nella regione Stati Uniti orientali (Ohio) (`us-east-2`). Nota che la regione è specificata nel valore ARN di`aws:SourceArn`, ma il valore dell'ID del server è un carattere jolly (*).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "opsworks-cm.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:opsworks-cm:us-east-2:123456789012:server/*"
        }
      }
    }
  ]
}
```

Esempio: aggiunta di più di un ARN del server a **`aws:SourceArn`**

L'esempio seguente limita l'accesso a una matrice di due AWS OpsWorks CM server nell'account ID 123456789012.

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "opsworks-cm.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnEquals": {
          "aws:SourceArn": [
             "arn:aws:opsworks-cm:us-east-2:123456789012:server/my-chef-
server/unique_ID",
             "arn:aws:opsworks-cm:us-east-2:123456789012:server/my-puppet-
server/unique_ID"
           ]
       }
      }
    }
  ]
}
```

Riservatezza del traffico Internet

AWS OpsWorks CM utilizza gli stessi protocolli di sicurezza della trasmissione generalmente utilizzati da AWS: HTTPS o HTTP con crittografia TLS.

Registrazione e monitoraggio in CM AWS OpsWorks

AWS OpsWorks CM registra tutte le azioni API su. CloudTrail Per ulteriori informazioni, consulta i seguenti argomenti:

- Registrazione OpsWorks per le chiamate API Puppet Enterprise con AWS CloudTrail
- Registrazione delle chiamate AWS OpsWorks for Chef Automate API con AWS CloudTrail

Convalida della conformità per AWS OpsWorks CM

AWS OpsWorks CM supporta i seguenti programmi e normative di conformità:

- Payment Card Industry (PCI)
- Health Insurance Portability and Accountability Act del 1996 (HIPAA)
- AWS Controlli di sistema e organizzazione (SOC) 1, 2 e 3
- Regolamento generale sulla protezione dei dati (GDPR)

I revisori esterni valutano la sicurezza e la conformità di AWS OpsWorks CM nell'ambito di molteplici programmi di AWS conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per un elenco di AWS servizi nell'ambito di programmi di conformità specifici, consulta AWS Services in Scope by Compliance Program. Per informazioni generali, consulta Programmi di conformità di AWS.

Puoi scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Download dei rapporti in AWS Artifact.

La vostra responsabilità di conformità quando utilizzate AWS OpsWorks CM è determinata dalla sensibilità dei vostri dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Guide Quick Start per la sicurezza e conformità: queste guide all'implementazione illustrano considerazioni relative all'architettura e forniscono fasi per l'implementazione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Whitepaper sull'architettura per la sicurezza e la conformità HIPAA: questo white paper descrive come le aziende possono utilizzare per creare applicazioni conformi allo standard HIPAA. AWS
- AWS Risorse per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe riguardare il settore e la località in cui operi.
- AWS Config— Questo AWS servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- AWS Security Hub— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente e consente AWS di verificare la conformità agli standard e alle best practice del settore della sicurezza.

Resilienza nel CM AWS OpsWorks

AWS OpsWorks CM abilita i backup giornalieri dei server per impostazione predefinita quando crei un server. I backup sono crittografati e archiviati in un bucket Amazon S3. Per impostazione predefinita,

questo bucket è accessibile solo all'account che ha creato il server. Puoi aggiungere l'accesso al bucket per altri utenti o configurare backup interregionali in Amazon S3 a tua discrezione. Chef e Puppet supportano la crittografia tra regioni perché entrambi i prodotti crittografano il traffico tra il server AWS OpsWorks CM e i nodi gestiti.

AWS OpsWorks CM non supporta configurazioni ad alta disponibilità (HA).

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, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture tradizionali a data center singolo o multiplo.

Per ulteriori informazioni su come eseguire il backup e il ripristino dei server in AWS OpsWorks CM, vedere quanto segue:

- Esegui il backup e il ripristino di un OpsWorks server Puppet Enterprise
- Backup e ripristino di un AWS OpsWorks for Chef Automate server

Per ulteriori informazioni su AWS regioni e zone di disponibilità, vedere AWS Global Infrastructure.

Sicurezza dell'infrastruttura in AWS OpsWorks CM

In quanto servizio gestito, AWS OpsWorks Configuration Management è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere a OpsWorks CM attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

AWS OpsWorks CM non supporta link privati o endpoint privati VPC.

AWS OpsWorks CM non supporta politiche basate sulle risorse. Per ulteriori informazioni, consulta AWS Servizi che funzionano con IAM nella Guida per l'AWS Identity and Access Management utente.

Analisi della configurazione e delle vulnerabilità in CM AWS OpsWorks

AWS OpsWorks CM esegue aggiornamenti periodici del kernel e della sicurezza del sistema operativo in esecuzione sul AWS OpsWorks server CM. Gli utenti possono impostare un intervallo di tempo per l'esecuzione degli aggiornamenti automatici per un massimo di due settimane dalla data corrente. AWS OpsWorks CM invia aggiornamenti automatici delle versioni secondarie di Chef e Puppet Enterprise. Per ulteriori informazioni sulla configurazione degli aggiornamenti per AWS OpsWorks for Chef Automate, consulta System Maintenance (Chef) in questa guida. Per ulteriori informazioni sulla configurazione degli aggiornamenti OpsWorks per Puppet Enterprise, consulta System Maintenance (Puppet) in questa guida.

Best practice di sicurezza per CM AWS OpsWorks

AWS OpsWorks CM, come tutti i AWS servizi, offre funzionalità di sicurezza da considerare durante lo sviluppo e l'implementazione delle proprie politiche di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per l'ambiente, gestiscile come considerazioni utili anziché prescrizioni.

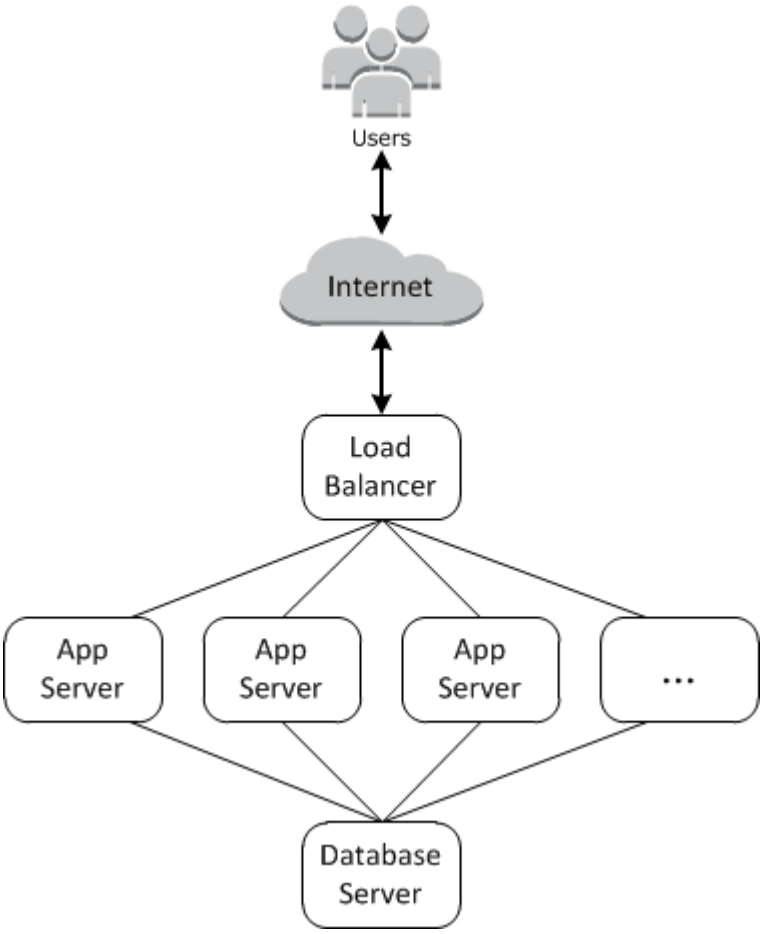
- Proteggi il tuo Starter Kit e le credenziali di accesso scaricate. Quando crei un nuovo server AWS OpsWorks CM o scarichi un nuovo Starter Kit e credenziali dalla console AWS OpsWorks CM, archivia questi elementi in un luogo sicuro che richieda almeno un fattore di autenticazione. Le credenziali forniscono l'accesso a livello di amministratore al server.
- Proteggi il tuo codice di configurazione. Proteggi il tuo codice di configurazione Chef o Puppet (libri di ricette e moduli) utilizzando i protocolli consigliati per i tuoi repository di origine. Ad esempio, è possibile limitare le autorizzazioni agli archivi o seguire le linee guida sul GitHub sito Web per proteggere gli archivi. AWS CodeCommit GitHub

- Utilizza i certificati firmati dall'autorità di certificazione per la connessione ai nodi. Sebbene sia possibile utilizzare certificati autofirmati durante la registrazione o l'avvio di nodi sul server AWS OpsWorks CM, come best practice è consigliabile utilizzare certificati firmati da un'autorità di certificazione. Si consiglia di utilizzare un certificato firmato di un'autorità di certificazione (CA).
- Non condividere le credenziali di accesso alla console di gestione Chef o Puppet con altri utenti. Un amministratore deve creare utenti separati per ogni utente dei siti Web della console Chef o Puppet.
  - Gestione degli utenti in Chef Automate
  - Gestione degli utenti in Puppet Enterprise
- Configura backup automatici e aggiornamenti di manutenzione del sistema. La configurazione degli aggiornamenti automatici di manutenzione sul server AWS OpsWorks CM garantisce che il server esegua gli aggiornamenti più recenti relativi alla sicurezza del sistema operativo. La configurazione dei backup automatici facilita il disaster recovery e velocizza i tempi di ripristino in caso di incidente o di errore. Limita l'accesso al bucket Amazon S3 che archivia i backup del server AWS OpsWorks CM; non concedere l'accesso a tutti. Concedi l'accesso in lettura o scrittura ad altri utenti singolarmente, se necessario, oppure crea un gruppo di sicurezza in IAM per tali utenti e assegna l'accesso al gruppo di sicurezza.
  - Manutenzione del sistema (Chef)
  - Manutenzione del sistema (Puppet)
  - Backup e ripristino di un AWS OpsWorks for Chef Automate server
  - Esegui il backup e il ripristino di un OpsWorks server Puppet Enterprise
  - Creazione del primo utente delegato e gruppo IAM nella Guida per l'utente di AWS Identity and Access Management
  - Best practice di sicurezza per Amazon S3 nella Amazon Simple Storage Service Developer Guide

AWS OpsWorks Pile

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

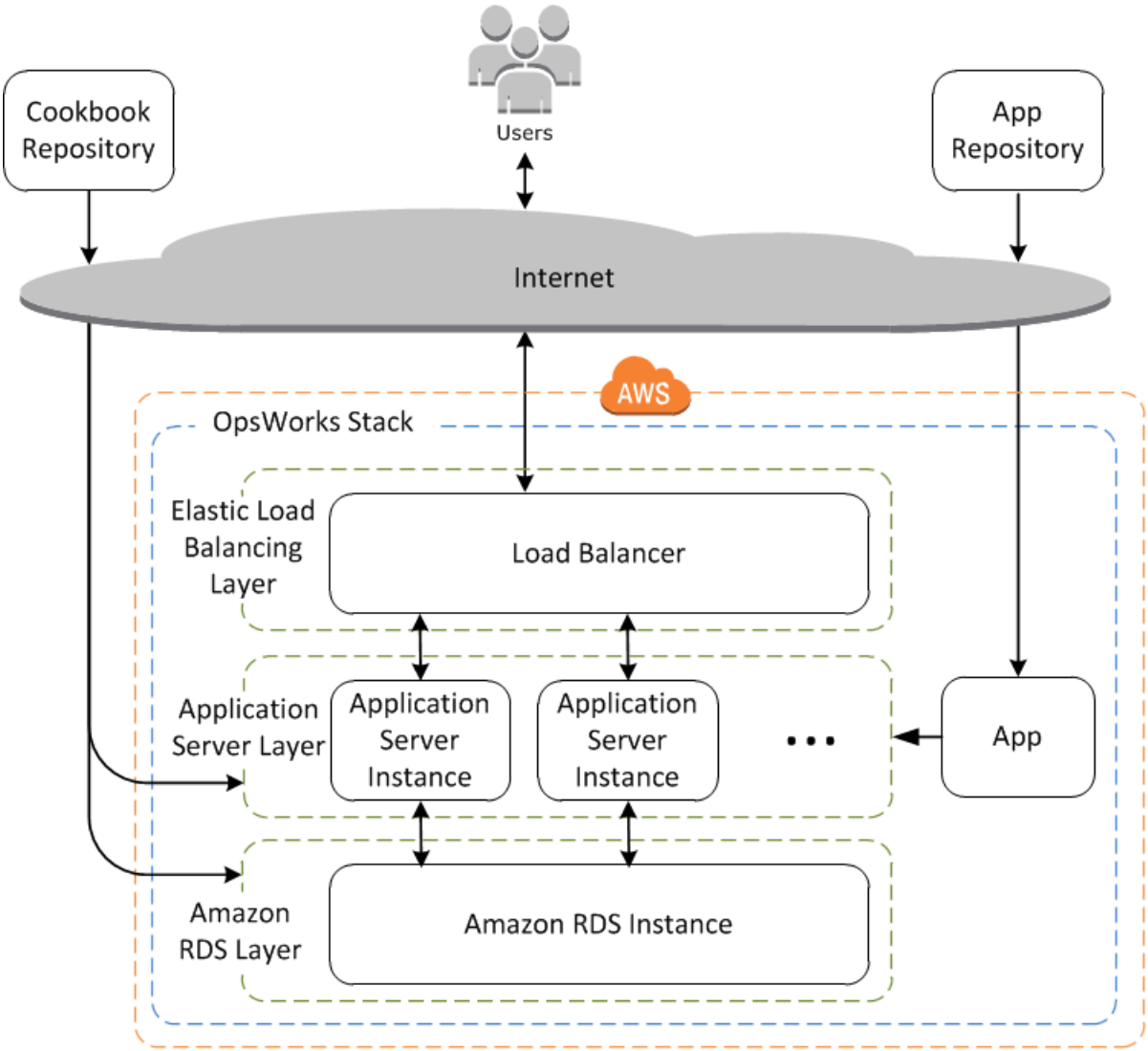
L'elaborazione basata sul cloud di solito coinvolge gruppi di risorse AWS, come istanze Amazon e EC2 istanze Amazon Relational Database Service (RDS), che devono essere create e gestite collettivamente. Ad esempio, un'applicazione Web richiede in genere server applicazioni, server di database, sistemi di bilanciamento del carico e così via. Questo gruppo di istanze viene in genere detto stack. Uno stack di un server applicazioni semplice può essere analogo al seguente.



Oltre a creare le istanze e installare i pacchetti necessari, è in genere necessario un modo per distribuire le applicazioni nei server applicazioni, monitorare le prestazioni dello stack, gestire la sicurezza e le autorizzazioni e così via.

AWS OpsWorks Stacks offre un modo semplice e flessibile per creare e gestire stack e applicazioni.

Ecco come potrebbe apparire uno stack di server applicativi di base con Stacks. AWS OpsWorks È costituito da un gruppo di server applicativi in esecuzione su un sistema di bilanciamento del carico Elastic Load Balancing, con un server di database Amazon RDS backend.



Sebbene sia relativamente semplice, questo stack mostra tutte le funzionalità principali di Stacks. AWS OpsWorks Ecco com'è composto.

Argomenti

- Stack

- Livelli
- Ricette ed eventi LifeCycle
- Istanze
- App
- Personalizzazione dello stack
- Gestione delle risorse
- Sicurezza e autorizzazioni
- Monitoraggio e registrazione
- CLI, SDK e modelli AWS CloudFormation
- AWS OpsWorks Stacks Fine del ciclo di vita FAQs
- Migrazione AWS OpsWorks Stacks delle applicazioni su AWS Systems Manager Application Manager
- Utilizzo dello strumento AWS OpsWorks Stacks Detach in Place
- Guida introduttiva a AWS OpsWorks Stacks
- AWS OpsWorks Le migliori pratiche di Stacks
- Stack
- Livelli
- Istanze
- App
- Libri di ricette e ricette
- Gestione delle risorse
- Tag
- Monitoraggio
- Sicurezza e autorizzazioni
- AWS OpsWorks Supporto Stacks per Chef 12 Linux
- Support per le versioni precedenti di Chef in AWS OpsWorks Stacks
- Utilizzo di AWS OpsWorks Stacks con altri servizi AWS
- Utilizzo della CLI AWS OpsWorks di Stacks
- Guida al debugging e alla risoluzione dei problemi
- AWS OpsWorks CLI di Stacks Agent
- AWS OpsWorks Riferimento a Stacks Data Bag

- OpsWorks Modifiche agli agenti

Stack

Lo stack è il componente principale AWS OpsWorks di Stacks. È fondamentalmente un contenitore per risorse AWS (istanze Amazon EC2 , istanze di database Amazon RDS e così via) che hanno uno scopo comune e devono essere gestite logicamente insieme. Lo stack ti aiuta a gestire queste risorse come un gruppo e definisce anche alcune impostazioni di configurazione predefinite, ad esempio il sistema operativo e la regione AWS delle istanze. Se desideri isolare alcuni componenti dello stack dall'interazione diretta dell'utente, puoi eseguire lo stack in un VPC.

Livelli

È possibile definire i componenti dello stack aggiungendo uno o più livelli. Un livello rappresenta un set di EC2 istanze Amazon che hanno uno scopo specifico, ad esempio rispondere alle richieste delle applicazioni o ospitare un server di database.

È possibile personalizzare o estendere i livelli modificando le configurazioni predefinite dei pacchetti, aggiungendo ricette Chef per eseguire attività come l'installazione di pacchetti aggiuntivi e molto altro.

Per tutti gli stack, AWS OpsWorks Stacks include livelli di servizio, che rappresentano i seguenti servizi AWS.

- Amazon Relational Database Service
- Sistema di bilanciamento del carico elastico
- Amazon Elastic Container Service

I livelli offrono un controllo completo sui pacchetti installati, sulla loro configurazione, sulla modalità di distribuzione delle applicazioni e molto altro.

Ricette ed eventi LifeCycle

I livelli dipendono dalle ricette Chef per gestire attività quali l'installazione di pacchetti nelle istanze, la distribuzione di app, l'esecuzione di script e così via. Una delle funzionalità principali di AWS OpsWorks Stacks è un insieme di eventi del ciclo di vita (Setup, Configure, Deploy, Undeploy e Shutdown) che eseguono automaticamente un set di ricette specificato al momento opportuno su ciascuna istanza.

Ogni livello può avere un set di ricette assegnato a ogni evento del ciclo di vita, che gestisce un'ampia gamma di attività per l'evento e il livello. Ad esempio, al termine dell'avvio di un'istanza appartenente a un livello di server Web, Stacks esegue le seguenti operazioni. AWS OpsWorks

1. Esegue le ricette dell'evento Setup (Impostazione) del livello, che possono eseguire attività come l'installazione e la configurazione di un server Web.
2. Esegue le ricette dell'evento Deploy (Distribuzione) del livello, che distribuiscono le applicazioni del livello da un repository all'istanza ed eseguono le attività correlate, ad esempio il riavvio del servizio.
3. Esegue le ricette dell'evento Configure (Configurazione) in ogni istanza nello stack, in modo che ogni istanza possa adattare la propria configurazione in base alla nuova istanza.

   Ad esempio, in un'istanza che esegue un sistema di bilanciamento del carico, una ricetta di Configure (Configurazione) può modificare la configurazione del sistema di bilanciamento del carico per includere la nuova istanza.

Se un'istanza appartiene a più livelli, AWS OpsWorks Stacks esegue le ricette per ogni livello in modo da poter, ad esempio, avere un'istanza che supporti un server di applicazioni PHP e un server di database MySQL.

Se hai implementato delle ricette, puoi assegnare ogni ricetta al livello e all'evento appropriati e AWS OpsWorks Stacks le eseguirà automaticamente al momento opportuno. Puoi anche eseguire le ricette manualmente, in qualsiasi momento.

Istanze

Un'istanza rappresenta una singola risorsa di elaborazione, ad esempio un' EC2 istanza Amazon. Definisce la configurazione di base della risorsa, ad esempio il sistema operativo e la dimensione. Altre impostazioni di configurazione, come gli indirizzi IP elastici o i volumi Amazon EBS, sono definite dai livelli dell'istanza. Le ricette del livello completano la configurazione eseguendo attività come l'installazione e la configurazione di pacchetti e la distribuzione di app.

Puoi usare AWS OpsWorks Stacks per creare istanze e aggiungerle a un livello. Quando avvii l'istanza, AWS OpsWorks Stacks avvia un' EC2 istanza Amazon utilizzando le impostazioni di configurazione specificate dall'istanza e dal suo livello. Al termine dell'avvio dell' EC2 istanza Amazon, AWS OpsWorks Stacks installa un agente che gestisce la comunicazione tra l'istanza e il servizio ed esegue le ricette appropriate in risposta agli eventi del ciclo di vita.

AWS OpsWorks Stacks supporta i seguenti tipi di istanze, caratterizzati dal modo in cui vengono avviate e interrotte.

- Le istanze 24 ore su 24, 7 giorni su 7 vengono avviate manualmente ed eseguite finché non vengono arrestate.
- Le istanze basate sul tempo vengono gestite da AWS OpsWorks Stacks secondo una pianificazione giornaliera e settimanale specificata.

  Permettono allo stack di regolare automaticamente il numero di istanze per supportare modelli di utilizzo prevedibili.
- Le istanze basate sul carico vengono avviate e interrotte automaticamente da AWS OpsWorks Stacks, in base a metriche di carico specifiche, come l'utilizzo della CPU.

  Permettono allo stack di regolare automaticamente il numero di istanze per supportare le variazioni nel traffico in ingresso. Le istanze basate sul carico sono disponibili solo per gli stack basati su Linux.

AWS OpsWorks Stacks supporta la riparazione automatica delle istanze. Se un agente interrompe la comunicazione con il servizio, AWS OpsWorks Stacks arresta e riavvia automaticamente l'istanza.

Puoi anche incorporare risorse di elaborazione basate su Linux in uno stack creato all'esterno di Stacks. AWS OpsWorks

- EC2 Istanze Amazon create direttamente utilizzando la EC2 console Amazon, la CLI o l'API.
- Istanze locali in esecuzione sul proprio hardware, incluse le istanze in esecuzione su macchine virtuali.

Dopo aver registrato una di queste istanze, diventa un'istanza AWS OpsWorks Stacks e puoi gestirla più o meno allo stesso modo delle istanze che crei con Stacks. AWS OpsWorks

App

Le applicazioni e i file correlati vengono archiviati in un repository, ad esempio un bucket Amazon S3. Ogni applicazione è rappresentata da un'app, che specifica il tipo di applicazione e contiene le informazioni necessarie per distribuire l'applicazione dal repository alle istanze, ad esempio l'URL e la password del repository. Quando distribuisci un'app, AWS OpsWorks Stacks attiva un evento Deploy, che esegue le ricette Deploy sulle istanze dello stack.

Puoi distribuire le app nei modi seguenti:

- Automaticamente: quando avvii le istanze, Stacks esegue automaticamente le ricette Deploy dell'istanza. AWS OpsWorks
- Manualmente: se hai una nuova app o desideri o aggiornarne una esistente, puoi eseguire manualmente le ricette dell'evento Deploy (Distribuzione) delle istanze online.

In genere, AWS OpsWorks Stacks esegue le ricette Deploy sull'intero stack, il che consente alle istanze degli altri livelli di modificare la propria configurazione in modo appropriato. Tuttavia, puoi limitare la distribuzione a un sottoinsieme di istanze se, ad esempio, desideri testare una nuova applicazione prima di distribuirla in ogni istanza del server applicazioni.

Personalizzazione dello stack

AWS OpsWorks Stacks offre vari modi per personalizzare i livelli e soddisfare i tuoi requisiti specifici:

- È possibile modificare il modo in cui AWS OpsWorks Stacks configura i pacchetti sovrascrivendo gli attributi che rappresentano le varie impostazioni di configurazione o persino sovrascrivendo i modelli utilizzati per creare i file di configurazione.
- È possibile estendere un livello esistente fornendo ricette personalizzate per eseguire attività quali l'esecuzione di script o l'installazione e la configurazione di pacchetti non standard.

Tutti gli stack possono includere uno o più livelli, che vengono avviati con solo un set minimo di ricette. Puoi aggiungere funzionalità al livello implementando ricette per gestire attività quali l'installazione di pacchetti, la distribuzione di app e così via. Imballa le tue ricette personalizzate e i file correlati in uno o più libri di cucina e archivia i libri di cucina in un repository come Amazon S3 o Git.

Puoi eseguire le ricette manualmente, ma AWS OpsWorks Stacks ti consente anche di automatizzare il processo supportando una serie di cinque eventi del ciclo di vita:

- Setup (Impostazione) si verifica in una nuova istanza dopo l'avvio.
- Configure (Configurazione) si verifica in tutte le istanze dello stack quando un'istanza passa allo stato online o esce da tale stato.
- Deploy (Distribuzione) si verifica quando un'app viene distribuita.
- Undeploy (Annullamento distribuzione) si verifica quando un'app viene eliminata.

- Shutdown (Arresto) si verifica quando un'istanza viene arrestata.

Ogni livello può avere un numero qualsiasi di ricette assegnate a ogni evento. Quando si verifica un evento del ciclo di vita sull'istanza di un layer, AWS OpsWorks Stacks esegue le ricette associate. Ad esempio, quando si verifica un evento Deploy su un'istanza dell'app server, AWS OpsWorks Stacks esegue le ricette Deploy del layer per scaricare l'app o eseguire attività correlate.

Gestione delle risorse

Puoi integrare altre risorse AWS, ad esempio indirizzi IP elastici, nello stack. Puoi utilizzare la console o l'API AWS OpsWorks Stacks per registrare risorse con uno stack, collegare risorse registrate o scollegarle dalle istanze e spostare le risorse da un'istanza all'altra.

Sicurezza e autorizzazioni

AWS OpsWorks Stacks si integra con AWS Identity and Access Management (IAM) per fornire metodi affidabili per controllare il modo in cui gli utenti accedono agli AWS OpsWorks stacks, tra cui:

- In che modo i singoli utenti possono interagire con ogni stack, ad esempio se possono creare risorse dello stack come livelli e istanze o se possono utilizzare SSH o RDP per connettersi alle istanze Amazon di uno stack. EC2
- In che modo AWS OpsWorks Stacks può agire per tuo conto per interagire con risorse AWS come le EC2 istanze Amazon.
- In che modo le app eseguite su istanze AWS OpsWorks Stacks possono accedere a risorse AWS come i bucket Amazon S3.
- Il modo in cui gestire le password RDP e le chiavi SSH pubbliche degli utenti e connettersi a un'istanza.

Monitoraggio e registrazione

AWS OpsWorks Stacks offre diverse funzionalità per aiutarti a monitorare lo stack e a risolvere i problemi relativi allo stack e a qualsiasi ricetta. Per tutti gli stack:

- AWS OpsWorks Stacks fornisce una serie di CloudWatch metriche personalizzate per gli stack Linux, riepilogate per comodità nella pagina Monitoraggio.

AWS OpsWorks Stacks supporta le metriche standard CloudWatch per gli stack di Windows. Puoi monitorarli con la console. CloudWatch

- CloudTrail log, che registrano le chiamate API effettuate da o per suo conto nel AWS OpsWorks tuo account AWS.
- Un log eventi che elenca tutti gli eventi nello stack.
- Log di Chef con indicazioni dettagliate di ciò che è accaduto per ogni evento del ciclo di vita in ogni istanza, ad esempio quali ricette sono state eseguite e quali errori si sono verificati.

Gli stack basati su Linux possono includere anche un master layer Ganglia, che puoi utilizzare per raccogliere e visualizzare dati di monitoraggio dettagliati per le istanze del tuo stack.

CLI, SDK e modelli AWS CloudFormation

Oltre alla console, AWS OpsWorks Stacks supporta anche un'interfaccia a riga di comando (CLI) e più lingue che possono essere utilizzate SDKs per eseguire qualsiasi operazione. Considera le seguenti caratteristiche:

- L'interfaccia a riga di comando di AWS OpsWorks Stacks fa parte dell'interfaccia a riga di comando di AWS e può essere utilizzata per eseguire qualsiasi operazione dalla riga di comando.

  L'interfaccia a riga di comando di AWS supporta più servizi AWS e può essere installata in Windows, Linux o sistemi OS X.
- AWS OpsWorks Stacks è incluso in AWS Tools per Windows PowerShell e può essere utilizzato per eseguire qualsiasi operazione da una riga di PowerShell comando di Windows.
- L'SDK AWS OpsWorks Stacks è incluso in SDKs AWS e può essere utilizzato da applicazioni implementate in: Java JavaScript(basato su browser e Node.js), .NET, PHP,Python (boto) o Ruby.

Puoi anche utilizzare modelli per effettuare il provisioning degli stack. AWS CloudFormation Per alcuni esempi, consulta AWS OpsWorks Snippets.

AWS OpsWorks Stacks Fine del ciclo di vita FAQs

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile.

Argomenti

- In che modo i clienti esistenti saranno influenzati da questa fine del ciclo di vita?
- AWS OpsWorks Stacks Accetta nuovi clienti?
- Dove devo migrare i miei stack esistenti?
- Come posso mantenere le mie EC2 istanze Amazon esistenti dopo la fine del ciclo di vita?
- La fine del ciclo di vita influirà su tutti Regioni AWS allo stesso tempo?
- Per quale livello di supporto tecnico è disponibile AWS OpsWorks Stacks?
- Saranno rilasciate nuove funzionalità per? AWS OpsWorks Stacks

In che modo i clienti esistenti saranno influenzati da questa fine del ciclo di vita?

I clienti esistenti non saranno interessati fino al 26 maggio 2024, data di fine del ciclo di vita del. AWS OpsWorks Stacks Dopo il 26 maggio 2024, i clienti non saranno più in grado di utilizzare la OpsWorks console, l'API, la CLI e le risorse. CloudFormation

AWS OpsWorks Stacks Accetta nuovi clienti?

No. AWS OpsWorks Stacks non accetta più nuovi clienti e al momento solo i clienti esistenti possono creare nuovi stack.

Dove devo migrare i miei stack esistenti?

Consigliamo AWS OpsWorks Stacks ai clienti di migrare i propri carichi di lavoro AWS Systems Manager dove possono sfruttare le seguenti funzionalità:

- Versioni Modern Chef
- SSM Agent
- Application Load Balancer
- Funzionalità di ridimensionamento avanzate tramite i gruppi Auto Scaling
- Capacità di definire le caratteristiche dell'host desiderate utilizzando EC2 modelli di lancio
- Tipi di istanze più recenti
- Tipi di volume EBS più recenti

Per informazioni su Systems Manager, vedere la Guida AWS Systems Manager per l'utente. Per informazioni sulla migrazione a AWS Systems Manager, vedere Migrazione AWS OpsWorks Stacks delle applicazioni su AWS Systems Manager Application Manager

Come posso mantenere le mie EC2 istanze Amazon esistenti dopo la fine del ciclo di vita?

Una volta raggiunta la data di fine del ciclo di vita, EC2 le istanze Amazon rimarranno nel tuo account, ma non potrai più utilizzare il servizio OpsWorks Stacks per controllare e gestire le istanze.

Puoi utilizzare lo strumento AWS OpsWorks Stacks Detach in Place per scollegare le OpsWorks istanze dal servizio Stacks. OpsWorks Dopo il distacco, puoi utilizzare Amazon EC2 o qualsiasi approccio EC2 compatibile per configurare e gestire le istanze. AWS Systems Manager Per ulteriori informazioni, consulta Utilizzo dello strumento AWS OpsWorks Stacks Detach in Place.

La fine del ciclo di vita influirà su tutti Regioni AWS allo stesso tempo?

Sì. La OpsWorks console, l'API, la CLI e CloudFormation le risorse verranno interrotte Regioni AWS contemporaneamente il 26 maggio 2024. Per un elenco delle aree Regioni AWS in cui AWS OpsWorks Stacks è disponibile, consulta l'elenco dei servizi AWS regionali.

Per quale livello di supporto tecnico è disponibile AWS OpsWorks Stacks?

AWS continuerà a fornire lo stesso livello di supporto AWS OpsWorks Stacks che i clienti hanno oggi fino alla data di fine del ciclo di vita. Se hai domande o dubbi, puoi contattare il Supporto AWS team su AWS re:POST o tramite Premium AWS Support.

Saranno rilasciate nuove funzionalità per? AWS OpsWorks Stacks

No. Poiché il servizio sta raggiungendo la fine del ciclo di vita, non rilasceremo alcuna nuova funzionalità. Tuttavia, continueremo a migliorare la sicurezza e a gestire le EC2 istanze Amazon come previsto fino alla data di fine del ciclo di vita.

Migrazione AWS OpsWorks Stacks delle applicazioni su AWS Systems Manager Application Manager

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile.

Ora puoi migrare AWS OpsWorks Stacks le tue applicazioni su Application Manager, una funzionalità che prevede l'utilizzo di uno AWS Systems Manager script di migrazione. La migrazione delle applicazioni Stacks a Systems Manager Application Manager consente di utilizzare AWS funzionalità che non sono disponibili in AWS OpsWorks Stacks, come nuovi tipi di EC2 istanze Amazon come Graviton, nuovi volumi Amazon Elastic Block Store (EBS) come gp3, nuovi sistemi operativi, integrazioni con gruppi Auto Scaling e sistemi di bilanciamento del carico delle applicazioni.

Con questa versione, è ora possibile monitorare ed eseguire operazioni sulle istanze migrate utilizzando una nuova scheda Istanze disponibile in Systems Manager Application Manager. È possibile utilizzare la scheda Istanze per visualizzare più AWS istanze in un'unica posizione. Utilizzando questa scheda, è possibile visualizzare informazioni sullo stato delle istanze e risolvere i problemi. Per ulteriori informazioni sull'utilizzo della scheda Istanze, consulta Lavorare con le istanze dell'applicazione nella Guida per l'utente.AWS Systems Manager

Argomenti

- Come funziona lo script
- Prerequisiti
- Limitazioni
- Nozioni di base

- Domande frequenti
- Risoluzione dei problemi

Come funziona lo script

AWS OpsWorks fornisce uno script che è possibile eseguire per migrare le AWS OpsWorks Stacks applicazioni a Systems Manager Application Manager utilizzando un CloudFormation modello.
Lo script ottiene informazioni su un OpsWorks livello esistente e, a seconda del valore del `--provision-application` parametro per lo script, fornisce un clone dell'applicazione o fornisce un CloudFormation modello iniziale che è possibile modificare utilizzando. AWS CloudFormation

Prerequisiti

- Assicuratevi che AWS CLI sia installato e configurato. Per ulteriori informazioni sull'installazione di AWS CLI, vedere Installazione o aggiornamento della versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente.

  > **Note**
  >
  > Se non si desidera configurare AWS CLI, è possibile eseguire i comandi anche utilizzando AWS CloudShell. Per ulteriori informazioni sull'utilizzo di CloudShell, vedere Working with AWS CloudShell nella Guida per l'AWS CloudShell utente.

- Assicurati che la versione 3.6 o successiva di Python sia installata o fornita con Amazon Machine Image (AMI).
- Assicurati che il tuo sistema operativo sia supportato. È possibile scaricare ed eseguire lo script di migrazione sui seguenti sistemi operativi.
  - Amazon Linux e Amazon Linux 2
  - Ubuntu 18.04 LTS, 20.04 LTS, 22.04 LTS
  - Red Hat Enterprise Linux 8
  - Windows Server 2019, Windows 10 Enterprise

  > **Note**
  >
  > Windows Server 2022 non è supportato.

Limitazioni

La nuova OpsWorks architettura è diversa dall'architettura per AWS OpsWorks Stacks. Questa sezione descrive i limiti noti di questa architettura.

Quanto segue non è supportato dalla nuova OpsWorks architettura.

- Esecuzione di ricette Chef su istanze Windows e CentOS
- Livelli Chef 11 e Berkshelf integrati
- Attributi e sacchetti di dati di Chef
- Istanze locali
- Istanze importate da EC2
- Nessun supporto per l'installazione di un elenco di pacchetti del sistema operativo specificato dall'utente
- Le app non sono supportate o migrate

Le seguenti sono supportate con limitazioni.

- Lo script di migrazione clona le informazioni sul volume EBS, ma esclude i punti di montaggio e i dati effettivi contenuti nei volumi.
- Le istanze scalate basate sul tempo e quelle basate sul carico vengono migrate, ma le regole di scalabilità associate a queste istanze non vengono migrate. È possibile modificare il gruppo Auto Scaling per ottenere risultati simili.
- Le entità IAM definite nella pagina Autorizzazioni dello stack nella OpsWorks console non vengono create o generate.
- Lo script di migrazione è in grado di eseguire il provisioning solo di applicazioni a livello singolo in Systems Manager. Ad esempio, se si esegue lo script due volte per due livelli nello stesso stack, si ottengono due applicazioni diverse in Systems Manager.

Nozioni di base

Lo script di migrazione,`stack_exporter.py`, è uno script Python che puoi eseguire localmente o su un' EC2 istanza. Prima di eseguire lo script, assicurati che tutti i prerequisiti siano soddisfatti. Per ulteriori informazioni sui prerequisiti, vedere. Prerequisiti

I passaggi nelle sezioni seguenti mostrano come migrare gli OpsWorks stack a Systems Manager Application Manager.

Argomenti

- Passaggio 1: Preparare l'ambiente per l'esecuzione dello script
- Passaggio 2: scarica lo script di migrazione
- Passaggio 3: configura l'ambiente per eseguire lo script
- Fase 4: Eseguire lo script
- Fase 5: Eseguire il provisioning di uno CloudFormation stack
- Fase 6: Esaminare le risorse disponibili
- Fase 7: Avviare un'istanza
- Passaggio 8: rivedi l'istanza
- Fase 9: Monitoraggio ed esecuzione delle operazioni sulle istanze utilizzando Systems Manager Application Manager

Passaggio 1: Preparare l'ambiente per l'esecuzione dello script

Prepara l'ambiente eseguendo i comandi appropriati per il tuo sistema operativo.

Argomenti

- Amazon Linux 2
- Amazon Linux
- Ubuntu 18.04, 20.04, 22.04
- Red Hat Enterprise Linux 8
- Windows Server 2019, Windows 10 Enterprise

Amazon Linux 2

```
sudo su
python3 -m pip install pipenv
PATH="$PATH:/usr/local/bin"
yum update
yum install git
```

Amazon Linux

```
sudo su
PATH="$PATH:/usr/local/bin"
export LC_ALL=en_US.utf-8
export LANG=en_US.utf-8
yum update
yum list | grep python3
yum install python36 // Any python version
yum install git
```

Per la versione 3.6 di Python, esegui anche:

```
python3 -m pip install pipenv==2022.4.8
```

Per Python versione 3.7 e successive, esegui anche:

```
python3 -m pip install pipenv
```

Ubuntu 18.04, 20.04, 22.04

```
sudo su
export PATH="${HOME}/.local/bin:$PATH"
apt-get update
apt install python3-pip
apt-get install git // if git is not installed
python3 -m pip install --user pipenv==2022.4.8
```

Red Hat Enterprise Linux 8

```
sudo su
sudo dnf install python3
PATH="$PATH:/usr/local/bin"
yum update
yum install git
python3 -m pip install pipenv==2022.4.8
```

Windows Server 2019, Windows 10 Enterprise

> **Note**
>
> Per Windows Server 2019, installa Python versione 3.6.1 o successiva.

```
pip install pipenv
```

Se Git non è già installato, scarica e installa Git.

Se usi Git come fonte per libri di cucina, aggiungi il tuo server Git a un `known_hosts` file prima di eseguire lo script su Windows. Puoi usare PowerShell per creare la seguente funzione.

```
function add_to_known_hosts($server){
    $new_host=$(ssh-keyscan $server 2> $null)
    $existing_hosts=''
    if (!(test-path "$env:userprofile\.ssh")) {
        md "$env:userprofile\.ssh"
    }
    if ((test-path "$env:userprofile\.ssh\known_hosts")) {
        $existing_hosts=Get-Content "$env:userprofile\.ssh\known_hosts"
    }
    $host_added=0
    foreach ($line in $new_host) {
        if (!($existing_hosts -contains $line)) {
            Add-Content -Path "$env:userprofile\.ssh\known_hosts" -Value $line
            $host_added=1
    }
   }
   if ($host_added) {
       echo "$server has been added to known_hosts."
   } else {
       echo "$server already exists in known_hosts."
   }
}
```

Puoi quindi fornire il tuo server Git (ad esempio, github.com, git-codecommit). *repository_region*.amazonaws.com) quando esegui la funzione.

```
add_to_known_hosts "myGitServer"
```

Passaggio 2: scarica lo script di migrazione

Scarica il file zip contenente lo script di migrazione e tutti i file pertinenti eseguendo il comando seguente.

```
aws s3api get-object \
    --bucket export-opsworks-stacks-bucket-prod-us-east-1 \
    --key export_opsworks_stacks_script.zip export_opsworks_stacks_script.zip
```

Se usi Linux, installa l'utilità unzip usando i seguenti comandi.

```
sudo apt-get install unzip
sudo yum install unzip
```

Decomprimi i file utilizzando il comando appropriato per il tuo sistema operativo.

Per Linux, usa il seguente comando.

```
unzip export_opsworks_stacks_script.zip
```

Per Windows, utilizzare il `Expand-Archive` comando in PowerShell.

```
Expand-Archive -LiteralPath PathToZipFile -DestinationPath PathToDestination
```

Dopo la decompressione del file, sono disponibili le cartelle e i file seguenti.

- README.md
- LICENZA
- NOTICE
- requirements.txt
- modelli/
  - OpsWorksCFNTemplate.yaml
  - Monta .yaml EBSVolumes
- opsworks/
- formazione di nuvole/
- tabula_istanze/
- cfn_stack_deployer.py

- s3.py
- stack_exporter_context.py
- stack_exporter.py

Passaggio 3: configura l'ambiente per eseguire lo script

Configura l'ambiente per eseguire lo script utilizzando il comando seguente.

```
pipenv install -r requirements.txt
pipenv shell
```

> **Note**
>
> Attualmente, lo script può eseguire il provisioning solo di applicazioni a livello singolo in Application Manager. Ad esempio, se si esegue lo script due volte per due livelli nello stesso stack, lo script crea due diverse applicazioni in Application Manager.

Dopo aver configurato l'ambiente, esaminate i parametri dello script. È possibile visualizzare le opzioni disponibili per lo script di migrazione eseguendo il `python3 stack_exporter.py --help` comando.

Parametro	Descrizione	Richiesto	Tipo	Valore predefinito
`--layer-id`	Esporta un CloudFormation modello per questo ID di OpsWorks livello.	Sì	string	
`--region`	La AWS regione dello OpsWorks stack. Se la regione OpsWorks dello stack e la regione dell'endp oint dell'API sono diverse, utilizza la regione dello stack. Questa è la stessa regione delle altre risorse che fanno parte dello OpsWorks stack (ad esempio, EC2 istanze e sottoreti).	No	string	us-east-1

Parametro	Descrizione	Richiesto	Tipo	Valore predefinito
`--provision-application`	Per impostazione predefinita, lo script esegue il provisioning dell'applicazione esportata dal modello. CloudFormation Passate questo parametro allo script con il valore FALSE per saltare il provisioning del modello. CloudFormation	No	Booleano	TRUE

Parametro	Descrizione	Richiesto	Tipo	Valore predefinito
`--launch-template`	Questo parametro definisce se utilizzare un modello di avvio esistente o creare un nuovo modello di lancio. È possibile creare un nuovo modello di avvio che utilizzi le proprietà dell'istanza consigliate o che utilizzi proprietà dell'istanza che corrispondono a un'istanza online. I valori validi includono: • `RECOMMENDED` - Utilizza le caratteristiche dell'istanza dell'AMI più recente per il sistema operativo OpsWorks dello stack e una dimensione dell'istanza c5.large. • `MATCH_LAST_INSTANCE` - Utilizza le caratteristiche delle istanze online più recenti disponibili. • *`LaunchTemplateID`* / [*`LaunchTemplateVersion]`* - Utilizza un modello di lancio esistente. Facoltativamente, puoi fornire una versione modello. Se non si fornisce una versione del modello, lo script utilizza la versione predefinita.	No	string	`RECOMMENDED`

Parametro	Descrizione	Richiesto	Tipo	Valore predefinito
`--system-updates`	Definisce se eseguire gli aggiornamenti del kernel e dei pacchetti all'avvio dell'istanza. I valori validi includono: • `ALL_UPDATES` - Esegue aggiornamenti di sistema per il kernel e i pacchetti all'avvio dell'istanza. • `NO_UPDATES` - Non esegue aggiornamenti di sistema all'avvio dell'istanza. • `MATCH_LAYER_SETTINGS` - Utilizza la `InstallUpdatesOnBoot` proprietà del OpsWorks livello o dell'istanza per determinare se installare gli aggiornamenti di sistema.	No	string	`ALL_UPDATES`
`--http-username`	Il nome del `SecureString` parametro Systems Manager che memorizza il nome utente utilizzato per l'autenticazione nell'archivio HTTP che contiene i libri di cucina personalizzati.	No	string	
`--http-password`	Il nome del `SecureString` parametro Systems Manager che memorizza la password utilizzata per l'autenticazione nell'archivio HTTP che contiene i libri di cucina personalizzati.	No	string	

Parametro	Descrizione	Richiesto	Tipo	Valore predefinito
`--repo-private-key`	Il nome del `SecureString` parametro Systems Manager che memorizza la chiave SSH utilizzata per l'autenticazione nel repository che contiene i ricettari personalizzati. Se il repository è attivo GitHub, è necessario generare una nuova chiave SSH. `Ed25519` Se non si genera una nuova chiave `Ed25519` SSH, la connessione al repository fallisce GitHub .	No	string	
`--lb-type`	L'eventuale tipo di bilanciamento del carico da creare durante la migrazione del sistema di bilanciamento del carico esistente . I valori validi includono: • `ALB`(Application Load Balancer) • `Classis`(Classic Load Balancer) • `None`(se non si desidera creare un sistema di bilanciamento del carico)	No	string	`ALB`

Parametro	Descrizione	Richiesto	Tipo	Valore predefinito
`--lb-access-logs-path`	Il percorso verso un bucket S3 esistente e il prefisso per l'archiviazione dei log di accesso al load balancer. Il bucket S3 e il load balancer devono trovarsi nella stessa regione. Se non si fornisce un valore e il valore del `--lb-type` parametro è impostato su`None`, lo script crea un nuovo bucket e prefisso S3. Assicurati che esista una policy bucket appropriata per questo prefisso.	No	string	
`--enable-instance-protection`	Se impostato su`TRUE`, lo script crea una politica di terminazione personalizzata (funzione Lambda) per il gruppo Auto Scaling. EC2le istanze con un `protected_instance` tag sono protette dagli eventi di scalabilità. Aggiungi un `protected_instance` tag a ogni EC2 istanza che desideri proteggere dagli eventi scalabili.	No	Booleano	`FALSE`
`--command-logs-bucket`	Il nome di un bucket S3 esistente per archiviare i log and. `AWS ApplyChefRecipe` `MountEBSVolumes` Se non fornisci un valore, lo script crea un nuovo bucket S3.	No	string	`aws-opsworks-application-manager-logs-`*`account-id`

Parametro	Descrizione	Richiesto	Tipo	Valore predefinito
`--custom-json-bucket`	Il nome di un bucket S3 esistente per memorizzare JSON personalizzato. Se non fornisci un valore, lo script crea un nuovo bucket S3.	No	string	`aws-apply-chef-application-manager-transition-data-`*`account-id`

Note:

- Se si utilizza un GitHub repository privato, è necessario creare una nuova chiave `Ed25519` host per SSH. Questo perché sono state GitHub modificate le chiavi supportate in SSH e rimosso il protocollo Git non crittografato. Per ulteriori informazioni sulla chiave `Ed25519` host, consulta il post sul GitHub blog Improving Git protocol security on GitHub. Dopo aver generato una nuova chiave `Ed25519` host, create un `SecureString` parametro Systems Manager per la chiave SSH e utilizzate il nome del `SecureString` parametro come valore per il `--repo-private-key` parametro. Per ulteriori informazioni su come creare un `SecureString` parametro Systems Manager, vedere Create a SecureString parameter (AWS CLI) o Create a Systems Manager (console) nella Guida per l'AWS Systems Manager utente.
- I `--http-username` `--repo-private-key` parametri `--http-password` e si riferiscono al nome di un `SecureString` parametro Systems Manager. Lo script di migrazione utilizza questi parametri quando si esegue il `AWS-ApplyChefRecipes` documento.
- Il `--http-username` parametro richiede che venga specificato anche un valore per il `--http-password` parametro.
- Il `--http-password` parametro richiede che venga specificato anche un valore per il `--http-username` parametro.
- Non impostate valori per entrambi `--http-password` e`--repo-private-key`. Fornire il nome del `SecureString` parametro Systems Manager di una chiave SSH (`--repo-private-key`) o il nome utente () e la password (`--http-username``--http-password`) del repository.

Fase 4: Eseguire lo script

Quando si eseguepython3 stack_exporter.py, è possibile eseguire il provisioning dell'applicazione o creare un modello iniziale impostando il valore del --provision-application parametro suFALSE.

Esempio 1: Eseguire il provisioning di un'applicazione Systems Manager Application Manager

Il comando seguente ottiene informazioni su un OpsWorks livello esistente ed effettua il provisioning di un'applicazione utilizzando l' OpsWorks architettura più recente, che consente di ottenere un risultato simile alla versione di Chef configurata per lo stack. Lo script fornisce tutte le risorse necessarie, ad esempio Auto Scaling group by using CloudFormation, e quindi registra l'applicazione in Systems Manager Application Manager.

Sostituisci *stack-region* e *layer-id* con i valori dello OpsWorks stack e del layer.

```
python3 stack_exporter.py \
     --layer-id layer-id \
     --region stack-region
```

Esempio 2: generare un modello

Il comando seguente ottiene informazioni su un OpsWorks livello esistente e genera un CloudFormation modello. Il modello, se fornito, ottiene un risultato simile all'utilizzo di Chef 14. In questo esempio, non viene fornita alcuna risorsa, poiché il --provision-application parametro è impostato su. FALSE

Sostituisci *stack-region* e *layer-id* con i valori OpsWorks dello stack e del layer.

```
python3 stack_exporter.py \
    --layer-id layer-id \
    --region stack-region \
    --provision-application FALSE
```

Dopo aver eseguito il comando, è possibile esaminare il modello nella libreria di modelli di Application Manager in Systems Manager ed è inoltre possibile eseguire il provisioning del modello. Per ulteriori informazioni sulla visualizzazione della libreria di modelli, vedere Lavorare con la libreria di modelli nella Guida per l'AWS Systems Manager utente.

Fase 5: Eseguire il provisioning di uno CloudFormation stack

> **Note**
>
> È necessario completare questo passaggio solo se si imposta il `--provision-application` parametro per lo script su`FALSE`.

Quando si specifica il `--provision-application` parametro con un valore di`FALSE`, l'output dello script fornisce il nome e l'URL del CloudFormation modello. Questo modello rappresenta una proposta sostitutiva dello OpsWorks stack e del layer esistenti.

È possibile eseguire il provisioning del modello utilizzando la libreria di modelli di Application Manager (consigliato) o utilizzando CloudFormation. Per ulteriori informazioni sull'utilizzo della libreria di modelli, vedere Lavorare con la libreria di modelli nella Guida per l'AWS Systems Manager utente.

Fase 6: Esaminare le risorse disponibili

Ora sei pronto per esaminare le risorse fornite.

1. Esamina le risorse per lo stack fornito utilizzando la console. AWS CloudFormation
   a. Apri la AWS CloudFormation console in https://console.aws.amazon.com/cloudformation e scegli Stacks.
   b. Nella pagina Stacks, scegli lo stack, quindi scegli la scheda Risorse.
   c. Nella scheda Risorse, esamina le risorse elencate per il tuo stack. L'elenco delle risorse include un gruppo EC2 Auto Scaling, che puoi consultare nella console Auto Scaling, oppure. AWS CLI
2. Esamina le risorse per l'applicazione utilizzando Systems Manager Application Manager.
   a. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
   b. Nel riquadro di navigazione, scegli Application Manager.
   c. Nella sezione Applicazioni, scegli l'applicazione personalizzata. Application Manager apre la scheda Panoramica.
   d. Scegliere la scheda Resources (Risorse). La scheda Risorse mostra tutte le risorse che sono state migrate per OpsWorks lo stack e il layer. Il nome dell'applicazione include il nome dello OpsWorks stack ed è formattato come *`app`* - *`stack-name`* -, *`suffix`* dove

suffix rappresenta i primi sei caratteri dell'ID dello stack. Per ulteriori informazioni sulla visualizzazione delle risorse in Application Manager, vedere Visualizzazione delle risorse dell'applicazione nella Guida per l'AWS Systems Manager utente.

Fase 7: Avviare un'istanza

Dopo aver effettuato il provisioning di un'istanza, sei pronto per testarla. A questo punto, non ci sono istanze in esecuzione.

Per portare le tue istanze online, modifica i `Min` `Desired capacity` valori e per il gruppo Auto Scaling a un numero adatto alla tua applicazione. `Max` Inizialmente, potresti voler impostare questi valori su 1 per portare online una singola istanza e verificare che l'istanza esegua tutte le azioni previste, inclusa l'esecuzione delle ricette Chef personalizzate.

Passaggio 8: rivedi l'istanza

Dopo aver avviato un'istanza, verifica che funzioni come previsto.

1. Esamina Chef `startup` e `terminate` i log che si trovano nel bucket S3 specificato dal parametro dello script. `--command-logs-bucket` Per impostazione predefinita, i log vengono archiviati in un bucket con il nome. `aws-opsworks-application-manager-logs-`*`account-id`*

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

   b. Scegli il bucket contenente i log.

   c. Vai al `ApplyChefRecipes` prefisso per visualizzare i log.

2. Verifica la connettività e lo stato dell'Application Load Balancer.

   Segui i passaggi seguenti per visualizzare i log di accesso per il tuo sistema di bilanciamento del carico. Puoi specificare il bucket S3 in cui desideri archiviare i log di accesso al load balancer utilizzando il parametro dello script. `--lb-access-logs-path`

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

   b. Scegli il tuo bucket S3, quindi vai al prefisso contenente i log.

3. Verifica che l'istanza superi tutti i controlli di integrità di Auto Scaling e Application Load Balancer (se ne hai configurati alcuni).

È possibile visualizzare le informazioni sullo stato di Auto Scaling nella nuova scheda Istanze.

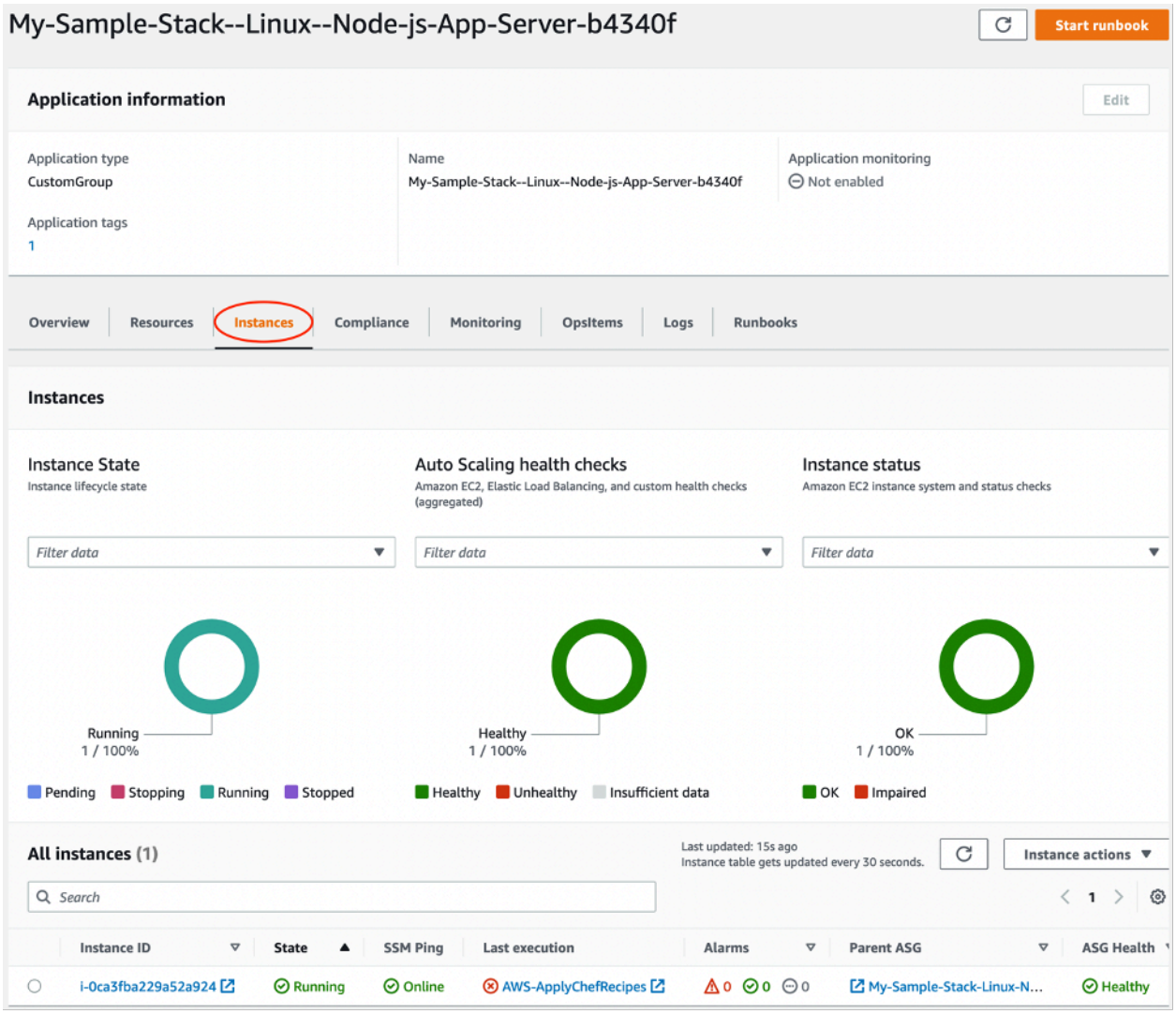
a. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
b. Nel riquadro di navigazione, scegli Application Manager.
c. Nella sezione Applicazioni, scegli Applicazioni personalizzate.
d. Scegli l'applicazione nell'elenco. Application Manager apre la scheda Panoramica.
e. Scegliete la scheda Istanze per visualizzare le informazioni sullo stato dell'Auto Scaling.

Dopo aver verificato che le ricette Chef vengano eseguite correttamente, puoi ridurre la capacità del gruppo Auto Scaling per terminare l'istanza. Se disponi di ricette di terminazione personalizzate, verifica che le ricette funzionino come previsto.

Fase 9: Monitoraggio ed esecuzione delle operazioni sulle istanze utilizzando Systems Manager Application Manager

È ora possibile monitorare ed eseguire operazioni sulle istanze utilizzando una nuova scheda Istanze nella pagina Application Manager. Per ulteriori informazioni sull'utilizzo della scheda Istanze, consulta Lavorare con le istanze dell'applicazione nella Guida per l'utente.AWS Systems Manager

È possibile utilizzare la scheda Istanze per visualizzare più AWS istanze in un'unica posizione. Utilizzando questa scheda, è possibile visualizzare informazioni sullo stato delle istanze e risolvere i problemi.



Effettua le seguenti operazioni per visualizzare la scheda Istanze.

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel riquadro di navigazione, scegli Application Manager.
3. Nella sezione Applicazioni, scegli Applicazioni personalizzate.
4. Scegli l'applicazione nell'elenco. Application Manager apre la scheda Panoramica.
5. Scegli la scheda Istanze per visualizzare le informazioni sullo stato e EC2 sullo stato dell'istanza.

Domande frequenti

Di seguito vengono FAQs fornite le risposte ad alcune domande comuni.

Argomenti

- Quali AWS OpsWorks Stacks versioni posso migrare?
- Quali versioni di Chef possono utilizzare le mie istanze migrate?
- Quali tipi di repository posso migrare?
- Posso continuare a usare un repository Git privato?
- Quali chiavi SSH posso usare per accedere alle mie istanze?
- Perché le mie istanze vengono ridimensionate automaticamente in avanti e indietro?
- Posso disattivare l'Auto Scaling?
- Posso eseguire aggiornamenti del kernel e dei pacchetti sulle istanze avviate EC2?
- Perché i volumi EBS nelle mie istanze non contengono dati?
- Perché i volumi EBS descritti nel mio modello di lancio non sono montati?
- Dove posso trovare i registri dei volumi Chef Recipe e Mount EBS?
- Dove posso trovare il registro di debug per lo script di migrazione?
- Lo script di migrazione supporta il controllo delle versioni dei modelli? CloudFormation
- Posso migrare più livelli?
- Come posso creare un SecureString parametro?
- Come posso proteggere le istanze del nuovo gruppo Auto Scaling dagli eventi di terminazione?
- Quali sistemi di bilanciamento del carico sono disponibili con lo script di migrazione?
- Le ricette personalizzate configurate per i libri di cucina sono state migrate?
- Posso eseguire le ricette di distribuzione e disinstallazione sulle mie istanze appena create?
- Posso modificare le sottoreti su cui si estende il mio gruppo Auto Scaling?

Quali AWS OpsWorks Stacks versioni posso migrare?

Puoi migrare solo gli stack Chef 11.10 e Chef 12, Amazon Linux, Amazon Linux 2, Ubuntu e Red Hat Enterprise Linux 7.

Quali versioni di Chef possono utilizzare le mie istanze migrate?

Le istanze migrate possono utilizzare le versioni di Chef dalla 11 alla 14.

> **Note**
>
> La migrazione dello stack di Windows non è supportata.

Quali tipi di repository posso migrare?

Puoi migrare i tipi di repository S3, Git e HTTP.

Posso continuare a usare un repository Git privato?

Sì, puoi continuare a utilizzare un repository Git privato.

Se si utilizza un GitHub repository privato, è necessario creare una nuova chiave `Ed25519` host per SSH. Questo perché sono state GitHub modificate le chiavi supportate in SSH e rimosso il protocollo Git non crittografato. Per ulteriori informazioni sulla chiave `Ed25519` host, consulta il post sul GitHub blog Improving Git protocol security on GitHub. Dopo aver generato una nuova chiave `Ed25519` host, create un `SecureString` parametro Systems Manager per questa chiave SSH e utilizzate il nome del parametro come valore per il `--repo-private-key` parametro. Per ulteriori informazioni su come creare un `SecureString` parametro Systems Manager, vedere Create a SecureString parameter (AWS CLI) nella Guida per l'AWS Systems Manager utente.

Per qualsiasi altro tipo di repository Git, crea un `SecureString` parametro Systems Manager per questa chiave SSH e usa il nome del parametro come valore per il parametro dello `--repo-private-key` script.

Quali chiavi SSH posso usare per accedere alle mie istanze?

Quando si esegue lo script, lo script migra le chiavi e le istanze SSH configurate nello stack. Puoi usare le chiavi SSH per accedere alla tua istanza. Se vengono fornite chiavi SSH per lo stack e l'istanza, lo script utilizza le chiavi dello stack. Se non sei sicuro di quali chiavi SSH usare, visualizza le istanze nella console (). EC2 https://console.aws.amazon.com/ec2/ La pagina Dettagli della EC2 console mostra le chiavi SSH per l'istanza.

Perché le mie istanze vengono ridimensionate automaticamente in avanti e indietro?

Auto Scaling ridimensiona le istanze in base alle regole di ridimensionamento per il gruppo Auto Scaling. È possibile impostare i valori di capacità minima, massima e desiderata per il gruppo. Il gruppo Auto Scaling ridimensiona automaticamente la capacità di conseguenza quando si aggiornano questi valori.

Posso disattivare l'Auto Scaling?

È possibile disattivare Auto Scaling impostando i valori di capacità minima, massima e desiderata del gruppo Auto Scaling sullo stesso numero. Ad esempio, se desideri avere sempre dieci istanze, imposta i valori di capacità minima, massima e desiderata su dieci.

Posso eseguire aggiornamenti del kernel e dei pacchetti sulle istanze avviate EC2?

Per impostazione predefinita, gli aggiornamenti del kernel e dei pacchetti avvengono all'avvio dell' EC2 istanza. Utilizza i seguenti passaggi per eseguire aggiornamenti del kernel o dei pacchetti su un'istanza avviata EC2. Ad esempio, potresti voler applicare gli aggiornamenti dopo aver eseguito deploy o configure recipes.

1. Connect alla tua EC2 istanza.
2. Crea la seguente `perform_upgrade` funzione ed eseguila sulla tua istanza.

```
perform_upgrade() {
    #!/bin/bash
    if [ -e '/etc/system-release' ] || [ -e '/etc/redhat-release' ]; then
     sudo yum -y update
    elif [ -e '/etc/debian_version' ]; then
     sudo apt-get update
     sudo apt-get dist-upgrade -y
    fi
}
perform_upgrade
```

3. Dopo gli aggiornamenti del kernel e dei pacchetti, potrebbe essere necessario riavviare l' EC2istanza. Per verificare se è necessario un riavvio, create la seguente `reboot_if_required` funzione ed eseguitela sulla vostra istanza. EC2

```
reboot_if_required () {
 #!/bin/bash
 if [ -e '/etc/debian_version' ]; then
   if [ -f /var/run/reboot-required ]; then
     echo "reboot is required"
   else
     echo "reboot is not required"
   fi
 elif [ -e '/etc/system-release' ] || [ -e '/etc/redhat-release' ]; then
  export LC_CTYPE=en_US.UTF-8
```

```
  export LC_ALL=en_US.UTF-8
  LATEST_INSTALLED_KERNEL=`rpm -q --last kernel | perl -X -pe 's/^kernel-(\S+).*/
$1/' | head -1`
  CURRENTLY_USED_KERNEL=`uname -r`
  if [ "${LATEST_INSTALLED_KERNEL}" != "${CURRENTLY_USED_KERNEL}" ];then
     echo "reboot is required"
  else
     echo "reboot is not required"
  fi
 fi
}
reboot_if_required
```

4. Se `reboot_if_required` esegui i risultati in un `reboot is required` messaggio, riavvia l' EC2 istanza. Se ricevi un `reboot is not required` messaggio, non è necessario riavviare l' EC2 istanza.

Perché i volumi EBS nelle mie istanze non contengono dati?

Quando esegui lo script, lo script migra la configurazione dei volumi EBS, creando un'architettura sostitutiva per lo stack e il OpsWorks layer. Lo script non migra le istanze effettive o i dati contenuti nelle istanze. Lo script migra solo la configurazione dei volumi EBS a livello di livello e collega i volumi EBS vuoti alle istanze avviate. EC2

Segui i passaggi seguenti per estrarre i dati dai volumi EBS delle tue istanze precedenti.

1. Scatta un'istantanea dei volumi EBS delle tue istanze precedenti. Per ulteriori informazioni sulla creazione di uno snapshot, consulta Create Amazon EBS snapshot nella Amazon EC2 User Guide.
2. Crea un volume dalla tua istantanea. Per ulteriori informazioni sulla creazione di un volume da uno snapshot, consulta Create a volume from a snapshot nella Amazon EC2 User Guide.
3. Collega il volume che hai creato alle istanze. Per ulteriori informazioni sul collegamento di volumi, consulta Collegare un volume Amazon EBS a un'istanza nella Amazon EC2 User Guide.

Perché i volumi EBS descritti nel mio modello di lancio non sono montati?

Se fornisci un ID del modello di avvio per il `--launch-template` parametro con i volumi EBS, lo script collega i volumi EBS, ma non monta i volumi. Puoi montare i volumi EBS collegati eseguendo il `MountEBSVolumes` RunCommand documento creato dallo script per l'istanza lanciata. EC2

Se non impostate il `--launch-template` parametro, lo script crea un modello e quando il gruppo Auto Scaling lancia una nuova EC2 istanza, il gruppo Auto Scaling collega automaticamente i volumi EBS e quindi esegue il `SetupAutomation` comando per montare i volumi collegati ai punti di montaggio configurati nelle impostazioni del livello.

Dove posso trovare i registri dei volumi Chef Recipe e Mount EBS?

OpsWorks consegna i log a un bucket S3 che puoi specificare fornendo un valore per il parametro. `--command-logs-bucket` Il nome predefinito del bucket S3 ha il formato:. `aws-opsworks-stacks-application-manager-logs-`*`account-id`* I registri delle ricette di Chef sono memorizzati nel prefisso. `ApplyChefRecipes` I registri dei volumi Mount EBS sono memorizzati nel prefisso. `MountEBSVolumes` Tutti i livelli migrati da uno stack inviano i log allo stesso bucket S3.

> **Note**
>
> - La configurazione del ciclo di vita del bucket S3 include una regola per eliminare i log dopo 30 giorni. Se desideri conservare i log per più di 30 giorni, devi aggiornare la regola nella configurazione del ciclo di vita del bucket S3.
> - Attualmente, registra OpsWorks solo Chef e ricette. `setup` `terminate`

Dove posso trovare il registro di debug per lo script di migrazione?

Lo script inserisce i log di debug in un bucket denominato. `aws-opsworks-stacks-transition-logs-`*`account-id`* Puoi trovare i log di debug nella `migration_script` cartella del bucket S3 all'interno delle cartelle che corrispondono al nome del layer che hai migrato.

Lo script di migrazione supporta il controllo delle versioni dei modelli? CloudFormation

Lo script genera documenti di tipo Systems Manager CloudFormation che sostituiscono il livello o lo stack da migrare. Eseguendo nuovamente lo script, anche con gli stessi parametri, si esporta una nuova versione del modello di livello precedentemente esportato. Le versioni del modello sono archiviate nello stesso bucket S3 dei log degli script.

Posso migrare più livelli?

Il `--layer-id` parametro dello script passa in un unico livello. Per migrare più livelli, esegui nuovamente lo script e passane uno diverso. `--layer-id`

I livelli che fanno parte dello stesso OpsWorks stack sono elencati nella stessa applicazione in Application Manager.

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel riquadro di navigazione, scegli Application Manager.
3. Nella sezione Applicazioni, scegli Applicazioni personalizzate.
4. Scegliere la applicazione. Il nome dell'applicazione inizia con `app-`*`stack-name`*`-`*`first-six-characters-stack-id`*.
5. L'elemento di primo livello che inizia con app, mostra tutti i componenti che corrispondono al tuo OpsWorks stack. Ciò include i componenti corrispondenti al OpsWorks livello.
6. Scegliete il componente corrispondente al livello per visualizzare le risorse per il livello. I componenti che rappresentano i OpsWorks livelli sono visibili anche nella sezione Applicazioni personalizzate come applicazioni singole.

Come posso creare un **`SecureString`** parametro?

È possibile utilizzare Systems Manager per creare un `SecureString` parametro. Per ulteriori informazioni su come creare un `SecureString` parametro Systems Manager, vedere Create a SecureString parameter (AWS CLI) o Create a Systems Manager (console) nella Guida per l'AWS Systems Manager utente.

È necessario fornire un `SecureString` parametro come valore per i `--repo-private-key` parametri `--http-username``--http-password`, o.

Come posso proteggere le istanze del nuovo gruppo Auto Scaling dagli eventi di terminazione?

È possibile proteggere le istanze impostando il `--enable-instance-protection` parametro su `TRUE` e aggiungendo una chiave di `protected_instance` tag a ciascuna EC2 istanza che si desidera proteggere dagli eventi di terminazione. Quando impostate il `--enable-instance-protection` parametro su `TRUE` e aggiungete una chiave `protected_instance` tag, lo script aggiunge una politica di terminazione personalizzata al nuovo gruppo Auto Scaling e sospende `ReplaceUnhealthy` il processo. Le istanze con la chiave `protected_instance` tag sono protette dai seguenti eventi di terminazione:

- Scalabilità degli eventi

- Aggiornamento istanza
- Ribilanciamento
- Durata massima dell'istanza
- Consenti la chiusura dell'istanza di quotazione
- Chiusura e sostituzione delle istanze non integre

> **Note**
>
> È necessario impostare la chiave del `protected_instance` tag sulle istanze che si desidera proteggere. La chiave tag fa distinzione tra maiuscole e minuscole. Qualsiasi istanza con quella chiave di tag è protetta indipendentemente dal valore del tag.
> Per ridurre il tempo di esecuzione della politica di terminazione personalizzata, puoi aumentare il numero predefinito di istanze utilizzate dalla funzione Lambda per filtrare le istanze protette aggiornando il valore della variabile del codice della funzione. `default_sample_size` Il valore predefinito è 15. Se si aumenta il`default_sample_size`, potrebbe essere necessario aumentare la memoria allocata alla funzione Lambda, il che aumenterebbe il costo della funzione Lambda. Per informazioni sui prezzi di AWS Lambda , consulta la pagina dei prezzi di AWS Lambda.

Quali sistemi di bilanciamento del carico sono disponibili con lo script di migrazione?

Lo script fornisce tre opzioni di bilanciamento del carico.

- (Consigliato) Creare un nuovo Application Load Balancer. Per impostazione predefinita, lo script crea un nuovo Application Load Balancer. È inoltre possibile impostare il `--lb-type` parametro su. `ALB` Per ulteriori informazioni sugli Application Load Balancer, vedi Cos'è un Application Load Balancer? nella Elastic Load Balancing User Guide.
- Se un Application Load Balancer non è un'opzione, crea un Classic Load Balancer `--lb-type` impostando il parametro su. `Classic` Se si seleziona questa opzione, il Classic Load Balancer esistente collegato al OpsWorks layer viene tenuto separato dall'applicazione. Per ulteriori informazioni sugli Application Load Balancer, vedi Cos'è un Classic Load Balancer? nella Guida per l'utente di Elastic Load Balancing: Classic Load Balancers.
- È possibile collegare un sistema di bilanciamento del carico esistente impostando il parametro su. `--lb-type None`

> **Important**
>
> Ti consigliamo di creare nuovi sistemi di bilanciamento del carico Elastic Load Balancing per i tuoi layer AWS Stacks. OpsWorks Se scegli di utilizzare un sistema di bilanciamento del carico Elastic Load Balancing esistente, devi prima confermare che non viene utilizzato per altri scopi e che non ha istanze collegate. Dopo aver collegato il load balancer al layer, OpsWorks rimuove tutte le istanze esistenti e configura il load balancer per gestire solo le istanze del layer. Sebbene sia tecnicamente possibile utilizzare la console o l'API Elastic Load Balancing per modificare la configurazione di un load balancer dopo averlo collegato a un layer, non dovresti farlo; le modifiche non saranno permanenti.

Per collegare il OpsWorks layer load balancer esistente al gruppo Auto Scaling

1. Esegui lo script di migrazione con il `--lb-type` parametro impostato su. `None` Quando il valore è impostato su`None`, lo script non clona né crea un sistema di bilanciamento del carico.
2. Dopo che lo script ha distribuito lo CloudFormation stack, aggiorna i `Min` `Max` gruppi `Desired capacity` e i valori di Auto Scaling, quindi testa l'applicazione.
3. Scegli `Link to the template` come mostrato nell'output dello script. Se hai chiuso il terminale, segui questi passaggi per accedere al modello.
   a. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
   b. Nel riquadro di navigazione, scegli Application Manager.
   c. Scegli CloudFormation pile e poi scegli Libreria di modelli.
   d. Scegli Owned by me e individua il tuo modello.
4. Dal CloudFormation modello, scegli Modifica dal menu Azioni.
5. Aggiorna la `LabelBalancerNames` proprietà all'interno della sezione delle `ApplicationAsg` risorse del CloudFormation modello.

```
ApplicationAsg:
   DependsOn: CustomTerminationLambdaPermission
   Properties:
   #(other properties in ApplicationAsg to remain unchanged)
      LoadBalancerNames:
        - load-balancer-name
```

```
      HealthCheckType: ELB
```

6. Se desideri che il controllo dello stato delle istanze di gruppo Auto Scaling utilizzi anche il controllo dello stato del sistema di bilanciamento del carico, rimuovi la sezione seguente ed inserisci. `HealthCheckType ELB` Se hai bisogno solo di controlli EC2 sanitari, non è necessario modificare il modello.
7. Salvare le modifiche. Il salvataggio crea una nuova versione predefinita del modello. Se è la prima volta che esegui lo script per il layer e la prima volta che salvi le modifiche nella console, la versione più recente è 2.
8. Da Azioni, scegli Provision stack.
9. Conferma di voler utilizzare la versione predefinita del modello. Assicurati che sia selezionata l'opzione Seleziona uno stack esistente e scegli lo CloudFormation stack da aggiornare.
10. Scegli Avanti per ciascuna delle pagine successive fino a visualizzare la pagina Revisione e fornitura. Nella pagina Review and Provision, scegli entrambi. Riconosco che AWS CloudFormation potrebbero creare risorse IAM con nomi personalizzati e comprendo che le modifiche AWS CloudFormation al modello selezionato possono causare l'aggiornamento o la rimozione di AWS risorse esistenti.
11. Scegli Provision stack (Effettua il provisioning di stack).

Se devi ripristinare gli aggiornamenti, procedi nel seguente modo.

1. Scegli Azioni, quindi scegli Provision stack.
2. Scegli Scegli una delle versioni esistenti, quindi scegli la versione precedente del modello.
3. Scegli Seleziona uno stack esistente, quindi scegli lo CloudFormation stack da aggiornare.

Le ricette personalizzate configurate per i libri di cucina sono state migrate?

L'esecuzione di Configure Custom Cookbook non è supportata durante un evento di configurazione. Lo script migra ricette personalizzate per la configurazione dei libri di cucina e crea un runbook di Systems Manager Automation per te. Tuttavia, è necessario eseguire le ricette manualmente.

Segui i passaggi seguenti per eseguire le ricette di configurazione.

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.

2. Nel riquadro di navigazione, scegli Application Manager.
3. Nella sezione Applicazioni, scegli Applicazioni personalizzate.
4. Scegliere la applicazione. Il nome dell'applicazione inizia con `app-`*`stack-name`*.
5. Scegli Risorse, quindi scegli il runbook di configurazione.
6. Scegli Execute Automation.
7. Scegli l'istanza IDs per la quale desideri eseguire le ricette di configurazione, quindi scegli Esegui.

Posso eseguire le ricette di distribuzione e disinstallazione sulle mie istanze appena create?

Lo script può creare tre possibili runbook di automazione a seconda della configurazione del livello.

- Configurazione
- Configura
- Interruzione

Lo script può anche creare i seguenti parametri di Systems Manager che contengono valori di input per il `AWS-ApplyChefRecipes Run Command` documento.

- Configurazione
- Implementazione
- Configura
- Undeploy (Annulla distribuzione)
- Interruzione

Quando si verifica un evento di scalabilità orizzontale, il runbook di installazione Automation viene eseguito automaticamente. Ciò include la configurazione e la distribuzione di ricette personalizzate per libri di cucina a partire dal livello originale. OpsWorks Quando si verifica un evento di scalabilità integrata, il runbook Terminate Automation viene eseguito automaticamente. Il runbook terminate Automation contiene le ricette di spegnimento del livello originale. OpsWorks

Se desideri eseguire undeploy o configurare le ricette manualmente, procedi nel seguente modo.

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel riquadro di navigazione, scegli Application Manager.
3. Nella sezione Applicazioni, scegli Applicazioni personalizzate.
4. Scegliere la applicazione. Il nome dell'applicazione inizia con `app-`*`stack-name`*`-`*`first-six-characters-stack-id`*. Application Manager apre la scheda Panoramica.
5. Scegli Risorse, quindi scegli il runbook di configurazione dell'automazione.
6. Scegli Execute Automation.
7. Per il parametro di input del runbook `applyChefRecipesPropertiesParameter` Automation, fate riferimento al parametro Systems Manager corretto. Il nome del parametro Systems Manager segue il formato `/ApplyChefRecipes-Preset/`*`OpsWorks-stack-name`*`-`*`OpsWorks-layer-name`*`-`*`first-six-characters-stack-id`*`/`*`event`*, in cui *`event`* è il valore per `ConfigureDeploy`, o `Undeploy` dipende dalle ricette che si desidera eseguire.
8. Scegliete l'istanza IDs in cui desiderate eseguire le ricette e scegliete Esegui.

Posso modificare le sottoreti su cui si estende il mio gruppo Auto Scaling?

Per impostazione predefinita, il gruppo Auto Scaling si estende su tutte le sottoreti dello stack VPC. OpsWorks Per aggiornare le sottoreti da estendere, procedi nel seguente modo.

1. Scegli `Link to the template` come mostrato nell'output dello script. Se hai chiuso il terminale, segui questi passaggi per accedere al modello.
   a. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
   b. Nel riquadro di navigazione, scegli Application Manager.
   c. Scegli CloudFormation pile e poi scegli Libreria di modelli.
   d. Scegli Owned by me e individua il tuo modello.
2. Da Actions, scegli Provision stack.
3. Conferma di voler utilizzare il modello predefinito. Scegli Seleziona uno stack esistente, quindi scegli lo CloudFormation stack da aggiornare.

> **Note**
>
> Se hai eseguito lo script con il `--provision-application` parametro impostato su`FALSE`, devi creare un nuovo CloudFormation stack.

4. Per il `SubnetIDs` parametro, fornite un elenco separato da virgole della sottorete su cui desiderate IDs che il gruppo Auto Scaling si estenda.
5. Scegli Avanti fino a visualizzare la pagina di revisione e fornitura.
6. Nella pagina Review and Provision, scegli Riconosco che AWS CloudFormation potrebbe creare risorse IAM con nomi personalizzati e comprendo che le modifiche AWS CloudFormation al modello selezionato possono causare l'aggiornamento o la rimozione di AWS risorse esistenti.
7. Scegli Provision stack (Effettua il provisioning di stack).

Risoluzione dei problemi

Questa sezione contiene alcuni problemi comuni e le relative soluzioni suggerite.

Argomenti

- Il valore principale fornito non è valido
- Impossibile eliminare lo CloudFormation stack quando le istanze protette del gruppo Auto Scaling sono abilitate
- Errore di accesso negato quando si forniscono il bucket e il prefisso S3 esistenti

Il valore principale fornito non è valido

Problema: viene visualizzato un messaggio di errore che indica che il capitale fornito non è valido.

Causa: Ciò si verifica perché il gruppo Auto Scaling non ha un ruolo di servizio.

Soluzione: creare un gruppo Auto Scaling nella regione in cui si è verificato l'errore. La creazione di un gruppo Auto Scaling crea il ruolo collegato al servizio necessario per la politica di terminazione personalizzata.

Impossibile eliminare lo CloudFormation stack quando le istanze protette del gruppo Auto Scaling sono abilitate

Problema: il `--enable-instance-protection` parametro è impostato su `TRUE` e alcune EC2 istanze del gruppo Auto Scaling sono protette con la chiave `protected_instance` tag, che impedisce l'eliminazione completa AWS CloudFormation dello stack.

Causa: le EC2 istanze hanno una chiave `protected_instance` tag che le protegge dagli eventi di terminazione.

Soluzione: rimuovete la chiave del `protected_instance` tag dalle istanze. EC2 Ciò consente al gruppo Auto Scaling di ridimensionarsi. Dopo che il gruppo Auto Scaling si è ridimensionato, puoi eliminare lo stack. AWS CloudFormation

Errore di accesso negato quando si forniscono il bucket e il prefisso S3 esistenti

Problema: ricevi un `AccessDenied` errore quando fornisci un bucket e un prefisso S3 esistenti.

Causa: la policy sui bucket S3 non fornisce le autorizzazioni necessarie per inviare i log del load balancer al bucket.

Soluzione: aggiorna la policy del bucket S3 per consentire allo script di inviare i log di accesso del load balancer al bucket. Per ulteriori informazioni su come aggiornare la policy del bucket, consulta Abilitare i log di accesso per l'Application Load Balancer nella Elastic Load Balancing: Application Load Balancers User Guide.

Utilizzo dello strumento AWS OpsWorks Stacks Detach in Place

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile.

Questa sezione descrive come utilizzare lo strumento AWS OpsWorks Stacks Detach in Place per scollegare le OpsWorks istanze dal servizio Stacks. OpsWorks

Le istanze scollegate rimarranno nelle tue Account AWS, ma non potrai più gestirle utilizzando. OpsWorks Utilizzerai invece Amazon EC2 o qualsiasi approccio EC2 compatibile per configurare e gestire le istanze. AWS Systems Manager

Ad alto livello, il processo di distacco prevede i seguenti passaggi:

1. Lo strumento esegue controlli di convalida per garantire che le risorse siano pronte per il distacco.
2. Lo strumento esporta il codice JSON personalizzato dallo OpsWorks stack e lo archivia come oggetto in Amazon S3.
3. Lo strumento crea documenti di Systems Manager Automation che rappresentano ogni evento del ciclo di vita di OpsWorks Stacks.
4. Lo strumento crea un AWS Service Catalog AppRegistry catalogo per tutte le istanze che vengono scollegate e scollega tutti i sistemi di bilanciamento del carico Elastic Load Balancing (ELB) dai livelli. OpsWorks
5. Infine, lo strumento scollega e annulla la registrazione di altre risorse, tra cui le istanze di Amazon Relational Database Service (Amazon RDS).

Come funziona il processo

Lo strumento Detach In Place fornisce i seguenti 3 comandi e un'esperienza simile a una procedura guidata che guida l'utente attraverso una serie di passaggi per controllare e configurare le istanze prima di procedere al distacco del layer.

Comando	Descrizione
`handle-prerequisites`	Questo comando analizza se tutte le istanze di un layer sono idonee per il distacco e risolve i prerequisiti. Le istanze devono essere integre in OpsWorks, non possono avere autoscaler basati sul tempo o sul carico e devono avere la versione più recente dell' OpsWorks agente installata. Inoltre, il comando verifica se tutte le istanze dispongono delle autorizzazioni necessarie per supportare l'agente SSM e se è installat

Comando	Descrizione
	a la versione più recente dell'agente SSM. Il comando installerà l'agente SSM se non è presente e aggiornerà l'agente SSM se non utilizza la versione più recente. Il comando aggiungerà anche tutte le autorizzazioni necessarie.
`detach`	Questo comando rimuove tutte le OpsWorks istanze del livello specificato. Innanzitutto, il comando eseguirà un controllo dei prerequisiti per garantire che il livello sia idoneo al distacco. Se non volete risolvere i prerequisiti, avete la possibilità di forzare il distacco. Successivamente, il comando indicherà che tutti i tag aggiunti alle istanze tramite OpsWorks tag APIs o propagazione dei tag dai livelli e dagli stack verranno mantenuti. Puoi rimuovere uno qualsiasi di questi tag utilizzan do il comando pertinente una EC2 APIs volta completato il distacco. Quindi, il comando verificherà se si desidera esportare la configurazione relativa a Chef in parametri SSM. Se al layer è collegato un Classic Load Balancer, il comando chiederà se è possibile scollegare il load balancer per evitare tempi di inattività.

Comando	Descrizione
`cleanup`	Questo comando elimina tutte le entità dal tuo account. OpsWorks Chiuderà le istanze ed eliminerà tutti gli stack. Questo dovrebbe essere usato per le risorse che non sono più necessarie come ultimo passaggio per ripulire l'account. **Note** Si consiglia di eseguire la nuova configurazione per alcuni giorni prima di eseguire il `cleanup` comando. Ciò garantisce che tutte le configurazioni necessarie dallo stack siano prontamen te disponibili, se necessario.

Limitazioni

Lo scopo principale dello strumento Detach In Place è quello di scollegare in modo sicuro le istanze Stacks. OpsWorks Questa sezione riassume i limiti dello strumento.

- Agente SSM di Windows: se l'agente SSM non è installato sull'istanza, dovrai installarlo manualmente. Lo stesso vale se l'agente non è aggiornato alla versione più recente.
- Istanze Time/Load Auto Scaling: lo strumento di distacco non supporta le istanze con Auto Scaling abilitato. È necessario disabilitare l'Auto Scaling sulle istanze che si desidera scollegare.
- Autorizzazioni: lo strumento di distacco non crea o genera entità IAM specificate nella pagina Autorizzazioni della console. OpsWorks
- App: lo strumento di distacco non crea o genera app al di fuori di. OpsWorks

Nozioni di base

Fase 1: Verifica che i prerequisiti siano soddisfatti

Tutti e 3 i comandi dello strumento Detach In Place sono script Python, che possono essere eseguiti localmente, EC2 su istanza o utilizzando. AWS CloudShell

AWS CloudShell è una shell basata su browser che consente l'accesso da riga di comando alle risorse del gruppo selezionato. AWS Regione AWS AWS CloudShell viene preinstallato con gli strumenti più diffusi (come AWS CLI Python). Quando si utilizza AWS CloudShell, si utilizzano le stesse credenziali utilizzate per accedere alla console.

Questa procedura dettagliata presuppone che tu stia utilizzando. AWS CloudShell

Passaggio 2: scarica lo script

1. Scarica il file zip che contiene lo script di migrazione e tutti i file pertinenti eseguendo il seguente comando:

   ```
   aws s3api get-object \
   --bucket detach-in-place-bucket-prod-us-east-1 \
   --key detach_in_place_script.zip detach_in_place_script.zip
   ```

2. Decomprimi il file eseguendo il comando seguente.

   ```
   unzip detach_in_place_script.zip
   ```

   Dopo aver decompresso il file, sono disponibili i seguenti file:

   - README.md
   - LICENZA
   - NOTICE
   - requirements.txt
   - TODO.py

3. Se necessario, eseguire l'installazione `pipenv` eseguendo il comando seguente.

   ```
   pip install pipenv
   ```

Passaggio 3: Esegui lo script

Innanzitutto, configura l'ambiente in modo da poter eseguire lo script eseguendo i seguenti comandi.

```
pipenv install -r requirements.txt
pipenv shell
```

Quindi, rivedi i parametri dello script.

Comando	Parametro	Descrizione	Tipo	Obbligatorio	Predefinita
`handle-prerequisites`	`--layer-id`	L'ID del livello che desideri staccare.	Stringa	Sì	-
	`--region`	La regione dello OpsWorks stack. Se la regione OpsWorks dello stack e la regione dell'endpoint dell'API sono diverse, utilizza la regione dello stack. Questa è la stessa regione delle altre risorse che fanno parte dello OpsWorks stack (ad esempio, EC2 istanze e sottoreti).	Stringa	No	us-east-1
`detach`	`--layer-id`	ID del livello che desideri staccare.	Stringa	Sì	-
	`--batch-size`	Numero di istanze da staccare da un livello (ad esempio, 5).	Stringa	No	-
	`--region`	La regione dello stack. OpsWorks Se la regione OpsWorks dello stack e la regione dell'endpoint	Stringa	No	us-east-1

Comando	Parametro	Descrizione	Tipo	Obbligatorio	Predefinita
		dell'API sono diverse, utilizza la regione dello stack. Questa è la stessa regione delle altre risorse che fanno parte dello OpsWorks stack (ad esempio, EC2 istanze e sottoreti).			
`cleanup`	`--stack-id`	ID dello stack che desideri eliminare.	Stringa	No	Escludendosi a vicenda, è necessario specificare un ID di livello o un ID dello stack
	`--layer-id`	ID del layer che si desidera eliminare	Stringa	No	
	`--region`	La regione dello OpsWorks stack. Se la regione OpsWorks dello stack e la regione dell'endpoint dell'API sono diverse, utilizza la regione dello stack. Questa è la stessa regione delle altre risorse che fanno parte dello OpsWorks stack (ad esempio, EC2 istanze e sottoreti).	Stringa	No	us-east-1

È possibile visualizzare le opzioni disponibili per i `detach` `cleanup` comandi `handle-prerequisites` e eseguendo i comandi con l'opzione nel modo seguente`--help`:

```
python3 layer_detacher.py detach --help
python3 layer_detacher.py handle-prerequisites --help
python3 layer_detacher.py cleanup --help
```

Ora sei pronto per iniziare. Gli esempi seguenti mostrano come eseguire i comandi per diversi casi d'uso.

Esempi:

- Esempio 1: verifica se un livello soddisfa tutti i prerequisiti ed è idoneo per il distacco
- Esempio 2: scollega tutte le istanze di un livello
- Esempio 3: scollega tutte le istanze di un livello in batch
- Esempio 4: pulisci tutte le risorse per un livello ed elimina il livello
- Esempio 5: pulisci tutte le risorse di uno stack ed elimina lo stack

Esempio 1: verifica se un livello soddisfa tutti i prerequisiti ed è idoneo per il distacco

Il comando seguente legge le informazioni su un OpsWorks livello (e le istanze che include) e verifica se sono soddisfatti i seguenti prerequisiti:

- Tutte le istanze sono online.
- Non esistono istanze Load/Time Auto Scaling.
- Tutte le istanze dispongono dell'agente più recente. OpsWorks
- In tutte le istanze è installato e configurato l'agente SSM più recente.
- Tutte le istanze hanno una coppia di chiavi SSH.
- Ogni istanza appartiene esattamente a un livello.

```
python3 layer_detacher.py handle-prerequisites \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Esempio 2: scollega tutte le istanze di un livello

Il comando seguente eseguirà un'iterazione su tutte le istanze del layer, verificherà se le istanze soddisfano i prerequisiti e proverà a scollegare in parallelo tutte le istanze che soddisfano i prerequisiti. Se uno o più prerequisiti non vengono soddisfatti, il comando fornirà un'opzione di distacco forzato per le restanti istanze non conformi.

Prima di scollegare qualsiasi istanza, il comando:

1. Salva il codice JSON personalizzato e caricalo su S3.
2. Crea documenti SSM Automation per ogni evento OpsWorks del ciclo di vita del layer e carica i log di esecuzione per i documenti di automazione su S3.
3. Crea un' AppRegistry applicazione per tutte le istanze che verranno scollegate. All'applicazione è associato un Resource Group che contiene tutte le istanze e le risorse separate. Le risorse includono documenti SSM Automation e parametri SSM che contengono informazioni sugli eventi del ciclo di vita e sulle ricette personalizzate di Chef.
4. Stacca il Classic Load Balancer dal layer, se ne esiste uno.

Questo comando modificherà solo OpsWorks le risorse. Lo stato delle EC2 istanze rimarrà lo stesso.

```
python3 layer_detacher.py detach \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Esempio 3: scollega tutte le istanze di un livello in batch

Il comando seguente esegue la stessa operazione dell'esempio precedente. L'unica differenza è che separa le istanze in batch.

Questo comando modificherà solo le risorse. OpsWorks Lo stato delle EC2 istanze rimarrà lo stesso.

```
python3 layer_detacher.py detach \
--layer-id opsworks-layer-id \
--region opsworks-stack-region \
--batch-size 5
```

Esempio 4: pulisci tutte le risorse per un livello ed elimina il livello

Il comando seguente eseguirà un'iterazione su tutte le risorse di un livello e le eliminerà. Più in dettaglio, interromperà ed eliminerà tutte le istanze e scollegherà il sistema di bilanciamento del

carico OpsWorks e annullerà la registrazione delle istanze EC2, degli elastici e dei volumi di Amazon RDS. IPs Dopo aver ripulito le risorse, eliminerà il layer.

Questo comando eliminerà OpsWorks risorse e EC2 istanze. Se vuoi che le tue EC2 istanze rimangano intatte, usa il `detach` comando prima di usare il comando. `cleanup` In questo modo il `cleanup` comando eliminerà tutte le risorse rimanenti.

```
python3 layer_detacher.py cleanup \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Esempio 5: pulisci tutte le risorse di uno stack ed elimina lo stack

Il comando seguente eseguirà un'iterazione su tutti i livelli e quindi eseguirà un'iterazione sulle risorse di ogni livello. Per ogni livello, il comando interromperà ed eliminerà tutte le istanze in e EC2, scollegherà i sistemi di bilanciamento del carico OpsWorks e annullerà la registrazione delle istanze, degli elastici e dei volumi di Amazon RDS. IPs Quindi, il comando eliminerà il livello. Lo stesso processo verrà eseguito in ogni livello che appartiene a questo stack. Infine, dopo che tutti i livelli sono stati eliminati, lo stack verrà rimosso.

Questo comando eliminerà OpsWorks risorse e EC2 istanze. Se vuoi che le tue EC2 istanze rimangano intatte, usa il `detach` comando prima di usare il comando. `cleanup` In questo modo il `cleanup` comando eliminerà tutte le risorse rimanenti.

```
python3 layer_detacher.py cleanup \
--stack-id opsworks-stack-id \
--region opsworks-stack-region
```

Passaggio 4: Continua a utilizzare le tue risorse dopo esserti staccato da OpsWorks

Dopo aver eseguito il `detach` comando, lo strumento crea una nuova AWS Service Catalog AppRegistry applicazione corrispondente al livello separato. Il nome dell'applicazione segue il formato*layer-name*`---`*layer-id*. Aggiunge inoltre il `OpsWorksLayerId` tag per identificare in modo univoco l'applicazione corrispondente al livello distaccato.

Per aggiungere nuove AWS risorse a questa applicazione (ad esempio nuove EC2 istanze), puoi effettuare una delle seguenti operazioni:

1. Assegna un tag alla risorsa con il tag di applicazione univoco dell' AppRegistryapplicazione:

Chiave tag: `awsApplication`

Valore: `arn:aws:resource-groups:`*`region`*`:`*`account-id`*`:group/`*`application-name`*`/`*`application-id`*`>`

2. Eseguire associate-resourcecomando.

Inoltre, per ogni AppRegistry applicazione, viene creato un gruppo di risorse. Il Resource Group contiene i seguenti tag.

Chiave tag	Valore
`EnableAWSServiceCatalogAppRegistry`	`TRUE`
`aws:servicecatalog:applicationName`	*`application-name`*
`aws:servicecatalog:applicationId`	*`application-id`*
`aws:servicecatalog:applicationArn`	`arn:aws:servicecatalog:` *`region`*`:`*`account-id`* `:/applications/` *`application-id`*

Esecuzione di compiti dopo il distacco

La tabella seguente fornisce informazioni su come eseguire le attività dopo il distacco:

Attività	Descrizione
Esecuzione di eventi del ciclo di vita	Dopo aver eseguito il `detach` comando e se è stata selezionata l'opzione, lo script crea 5 documenti di automazione corrispondenti ai 5 eventi del OpsWorks ciclo di vita. Il nome di ogni documento di automazione segue questo formato:. *`layer-`*

Attività	Descrizione
	id_lifecycle-event _automati on_document Per simulare OpsWorks il comportamento in Systems Manager, è necessario attivare manualmente le esecuzioni di automazione durante il provisioning, la chiusura delle EC2 istanze o la distribuzione/rimozione di ricette.
Aggiornamento del codice JSON personali zzato	I codici JSON personalizzati per lo stack e il layer vengono archiviati in un bucket S3 specificato durante il distacco o, in alternativa, in un nuovo bucket S3 che viene creato. I nomi di file archiviati per i file JSON sono i seguenti: • layercustomjson.json • stackcustomjson.json
Modifica dell'elenco di esecuzione per gli eventi del ciclo di vita	L'elenco di esecuzione per ogni evento del ciclo di vita è definito nel documento di automazio ne corrispondente. Per modificare l'elenco delle esecuzioni, trova i documenti di automazione nell' AppRegistry applicazione e modifica il RunList parametro. Il processo di aggiornamento delle ricette e dei libri di cucina rimane invariato perchéAWS-ApplyChefRecipes , attivato dai documenti di Automation, supporta la stessa fonte di. OpsWorks

Attività	Descrizione
Gestione della riparazione automatica/scalabilità automatica	Quando si scollega un'istanza, l' OpsWorks agente viene disinstallato. Senza l'agente, OpsWorks non è possibile correggere o sostituire automaticamente le istanze non integre, né è possibile scalare automaticamente il parco veicoli. Per continuare a scalare automaticamente e sostituire le istanze fallite, crea un gruppo Amazon Auto EC2 Scaling. Il gruppo lancerà nuove istanze per mantenere la capacità desiderata quando Amazon EC2 rileva istanze non integre che devono essere sostituite.
Gestione del Load Balancer	Se il layer utilizza un Classic Load Balancer, il `detach` comando lo scollegherà prima di annullare la registrazione delle istanze. Questo viene fatto per garantire che tutte le associazioni di istanze ELB rimangano conservate su Amazon EC2 durante il processo di distacco, riducendo così a zero i tempi di inattività. Una volta completato il processo, sarai in grado di gestire il tuo ELB su. EC2

Attività	Descrizione
Connessione all'istanza	Quando si esegue il `detach` comando `handle-prerequisites` or, vengono eseguiti due controlli: • La versione dell'agente SSM e le autorizza zioni • Chiavi SSH I comandi offrono anche la possibilità di aggiornare l'agente SSM e aggiungere le autorizzazioni necessarie in modo da poterti connettere alle istanze utilizzando Session Manager. Se esistono chiavi SSH, hai anche la possibilità di inserire SSH nell'istanza.

Utilizzo della scheda Istanze di Systems Manager Application Manager

Dopo il distacco, sarete in grado di visualizzare e gestire le istanze nella scheda Istanze di Application Manager.

La scheda Istanze fornisce informazioni aggregate sulle EC2 istanze di un'applicazione, come lo stato, lo stato di integrità e lo stato dell'ultimo comando. Utilizzando questa scheda, è possibile visualizzare informazioni dettagliate sulle singole istanze come la cronologia dei comandi, gli stati di allarme, lo stato degli agenti di Systems Manager e altro ancora. La scheda Istanze offre anche una serie di azioni, come la possibilità di applicare ricette Chef, avviare o arrestare un'istanza o aggiungere o rimuovere un'istanza da un gruppo Auto Scaling.

Guida introduttiva a AWS OpsWorks Stacks

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks offre un ricco set di componenti personalizzabili che puoi combinare per creare uno stack che soddisfi i tuoi scopi specifici. La sfida per i nuovi utenti è comprendere come assemblare questi componenti in uno stack di lavoro e gestirlo in modo efficace. Scopri di seguito come iniziare.

Se vuoi...	Completa questa procedura guidata:
Creare uno stack di esempio più rapidamente possibile	Nozioni di base: esempio
Provare un stack basato su Linux	Nozioni di base: Linux
Provare un stack basato su Windows	Nozioni di base: Windows
Scoprire come creare il tuoi libri di ricette Chef	Nozioni di base: libri di ricette

Se disponi di risorse di elaborazione esistenti, ad EC2 esempio istanze Amazon o anche istanze locali in esecuzione sul tuo hardware, puoi incorporarle in uno stack, insieme alle istanze che hai creato con Stacks. AWS OpsWorks Puoi quindi utilizzare AWS OpsWorks Stacks per gestire tutte le istanze correlate come gruppo, indipendentemente da come sono state create.

Supporto nelle regioni

Puoi accedere a AWS OpsWorks Stacks a livello globale; puoi anche creare e gestire istanze a livello globale. Gli utenti possono configurare le istanze AWS OpsWorks Stacks in modo che vengano lanciate in qualsiasi AWS regione tranne AWS GovCloud (Stati Uniti occidentali) e la regione della Cina (Pechino). Per utilizzare AWS OpsWorks Stacks, le istanze devono essere in grado di connettersi a uno dei seguenti endpoint dell'API del servizio di istanze AWS OpsWorks Stacks.

Le risorse possono essere gestite solo nella regione in cui sono state create. Le risorse create in un endpoint regionale non sono disponibili e non possono essere clonate in un altro endpoint regionale. Puoi avviare istanze nelle seguenti regioni.

- Stati Uniti orientali (Ohio)

- Stati Uniti orientali (Virginia settentrionale)
- Regione Stati Uniti occidentali (Oregon)
- Regione Stati Uniti occidentali (California settentrionale)
- Regione Canada (centrale) (solo API, non disponibile per gli stack creati in AWS Management Console.)
- Regione Asia Pacifico (Mumbai)
- Regione Asia Pacifico (Singapore)
- Regione Asia Pacifico (Sydney)
- Regione Asia Pacifico (Tokyo)
- Regione Asia Pacifico (Seoul)
- Regione Europa (Francoforte)
- Regione Europa (Irlanda)
- Regione Europa (Londra)
- Regione Europa (Parigi)
- Regione Sud America (San Paolo)

Nozioni di base sugli stack di esempio

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa procedura dettagliata mostra come utilizzare AWS OpsWorks Stacks per creare rapidamente un ambiente applicativo Node.js di esempio con pochi clic del mouse e senza scrivere codice. Al termine, hai un'istanza Amazon Elastic Compute Cloud (Amazon EC2) che esegue Chef 12, un server HTTP Node.js e un'app Web che puoi utilizzare per interagire con Twitter e lasciare commenti su una pagina Web.

> **Note**
>
> Poiché il completamento di questa procedura dettagliata crea automaticamente un'istanza con un tipo di c3.large, non puoi utilizzare questa procedura dettagliata o lo strumento di creazione Sample Stack in Stacks, nel piano gratuito. AWS OpsWorksAWS Sebbene l'utilizzo dello strumento di creazione Sample Stack in un VPC crei un'istanza t2.medium VPCs , attualmente non sono disponibili nel piano gratuito.AWS

Fase 1: completamento dei prerequisiti

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Prima di poter avviare la procedura guidata, devi completare le seguenti fasi di configurazione. Questi passaggi di configurazione includono la registrazione di un AWS account, la creazione di un utente amministrativo e l'assegnazione delle autorizzazioni di accesso a Stacks. AWS OpsWorks

Argomenti

- Registrati per un Account AWS
- Crea un utente con accesso amministrativo
- Assegna le autorizzazioni di accesso al servizio

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

Parte della procedura di registrazione prevede la ricezione di una telefonata o di un messaggio di testo e l'immissione di un codice di verifica sulla tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Assegna le autorizzazioni di accesso al servizio

Abilita l'accesso al servizio AWS OpsWorks Stacks (e ai servizi correlati su cui si basa AWS OpsWorks Stacks) aggiungendo le `AmazonS3FullAccess` autorizzazioni `AWSOpsWorks_FullAccess` e al tuo ruolo o utente.

Per ulteriori informazioni sull'aggiunta di autorizzazioni, consulta Aggiungere i permessi di identità IAM (console).

A questo punto hai completato tutte le fasi di configurazione. Ora puoi avviare questa procedura guidata.

Fase 2. Creazione di uno stack

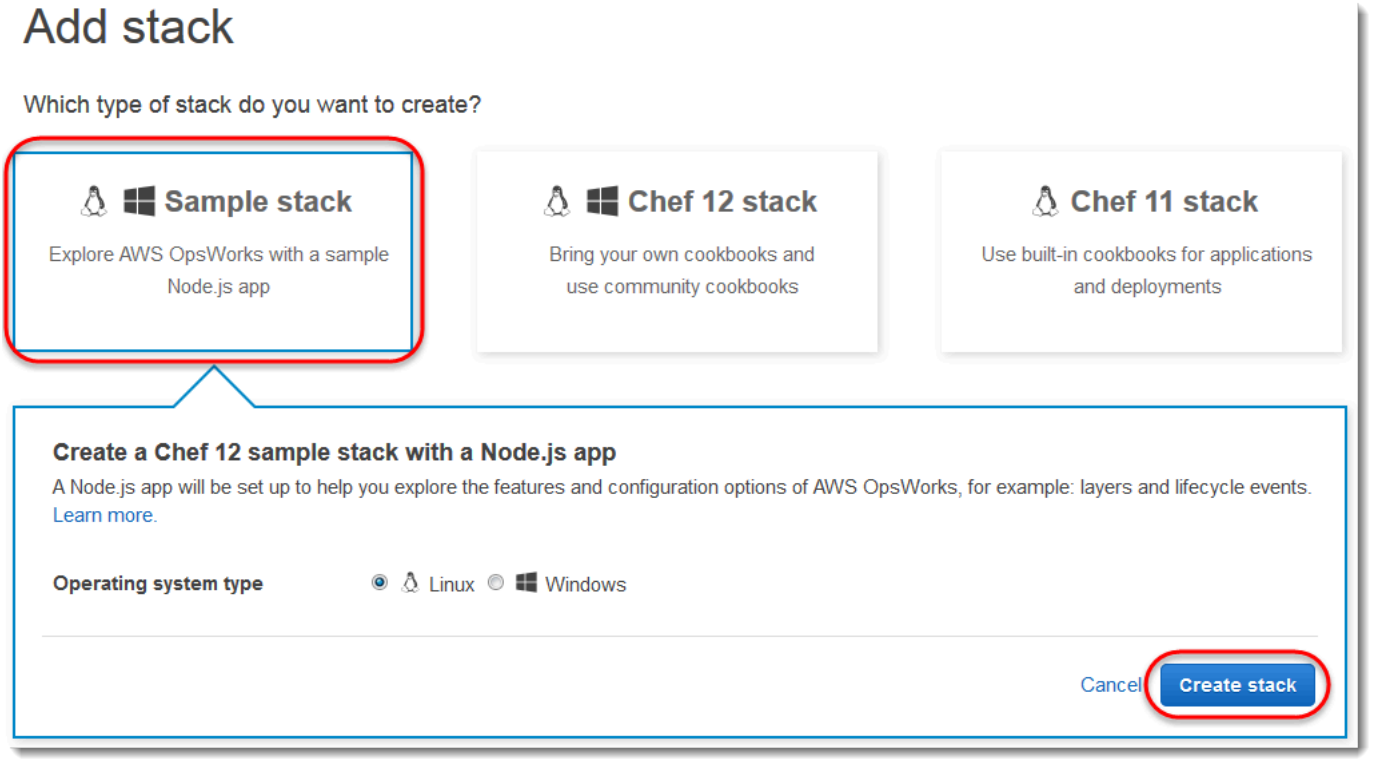
> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

In questo passaggio, usi la console AWS OpsWorks Stacks per creare uno stack. Uno stack è una raccolta di istanze (come le EC2 istanze Amazon) e AWS risorse correlate che hanno uno scopo comune e che si desidera gestire insieme. Per ulteriori informazioni, consulta Stack. In questa procedura guidata sarà disponibile solo un'istanza.

Prima di iniziare questa fase, completa i prerequisiti.

Per creare lo stack

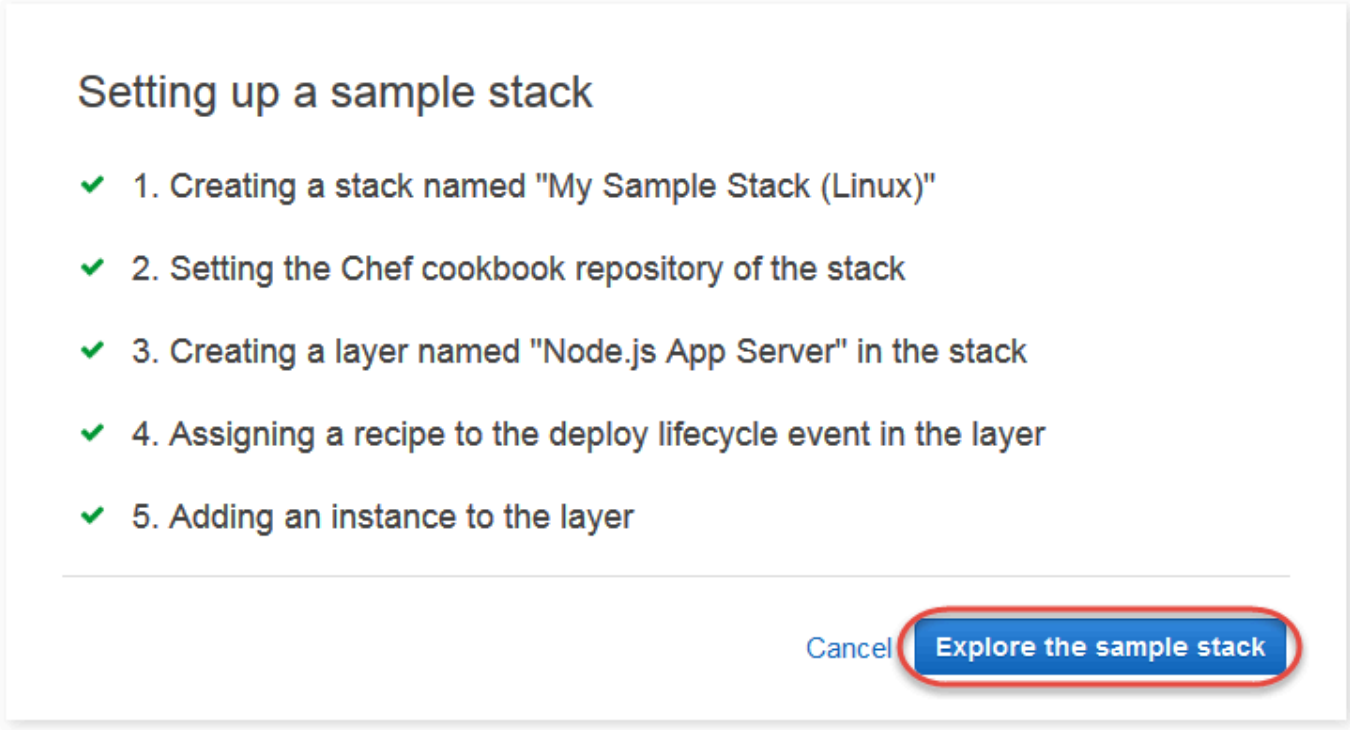
1. Accedi AWS Management Console e apri la console all'indirizzo. AWS OpsWorks https://console.aws.amazon.com/opsworks/
2. Eseguire una delle seguenti operazioni, se possibile:
   - Se viene visualizzata la pagina Welcome to AWS OpsWorks Stacks, scegli Aggiungi il tuo primo stack o Aggiungi il tuo primo AWS OpsWorks stack Stacks (entrambe le scelte hanno la stessa funzione). Viene visualizzata la pagina Add stack (Aggiungi stack).
   - Se viene visualizzata la pagina OpsWorks Dashboard, scegli Aggiungi stack. Viene visualizzata la pagina Add stack (Aggiungi stack).
3. Con la pagina Add stack (Aggiungi stack) visualizzata, scegliere Sample stack (Stack di esempio), se l'opzione non è già stata selezionata automaticamente.
4. Dopo aver selezionato Linux in Operating system type (Tipo di sistema operativo), scegliere Create stack (Crea stack):



5. AWS OpsWorks Stacks crea uno stack denominato My Sample Stack (Linux). AWS OpsWorks Stacks aggiunge anche tutti i componenti necessari per distribuire l'app nello stack:

   - Un livello, che è un modello per un set di istanze. Specifica elementi quali, ad esempio, le impostazioni dell'istanza, le risorse, i pacchetti installati e i gruppi di sicurezza. Per ulteriori informazioni, consulta Livelli. Il nome del livello è Node.js App Server (Server app Node.js).
   - Un'istanza, che in questo caso è un' EC2istanza di Amazon Linux 2. Per ulteriori informazioni sulle istanze, consulta Istanze. Il nome host dell'istanza è nodejs-server1.
   - Un'app, che è il codice da eseguire sull'istanza. Per ulteriori informazioni sulle app, consulta App. Il nome dell'app è Node.js Sample App (App di esempio Node.js).

6. Dopo che AWS OpsWorks Stacks ha creato lo stack, scegli Esplora lo stack di esempio per visualizzare la pagina My Sample Stack (Linux) (se completi questa procedura dettagliata più volte, My Sample Stack (Linux) potrebbe avere un numero sequenziale dopo di essa, ad esempio 2 o 3):



Nella fase successiva l'istanza verrà avviata e si procederà alla distribuzione dell'app sull'istanza.

Fase 3. Avvio dell'istanza e distribuzione dell'app

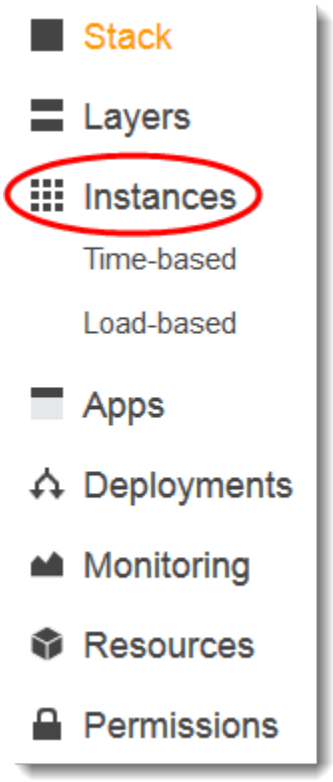
> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ora che disponi di un'istanza e di un'app, avvia l'istanza e distribuisci l'applicazione sull'istanza.

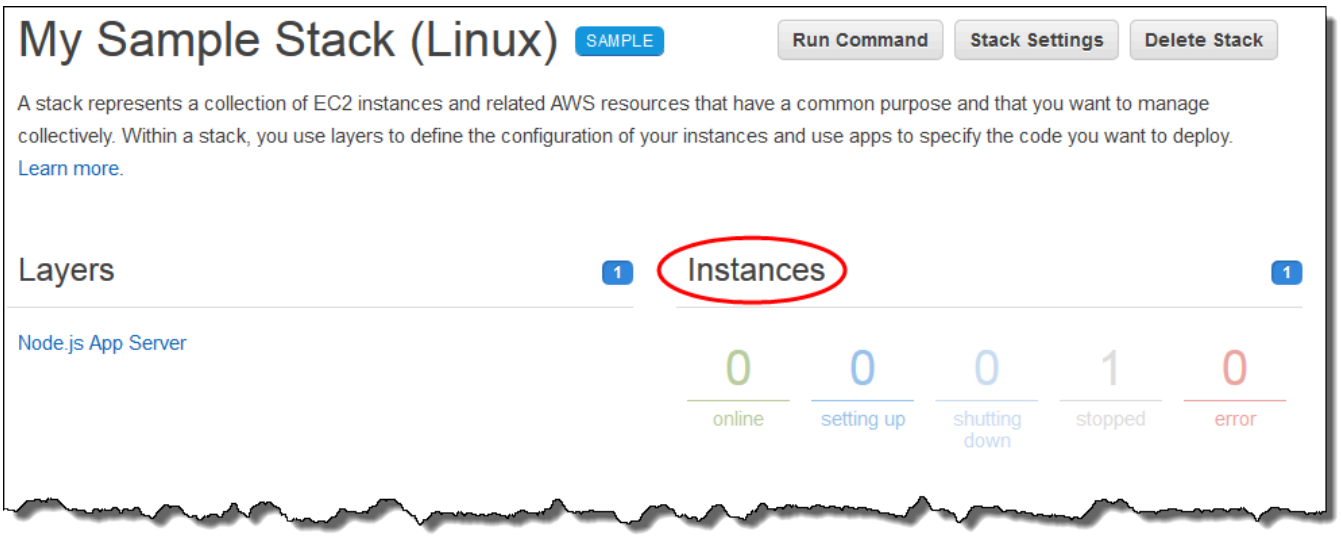
Per avviare l'istanza e distribuire l'app

1. Esegui una di queste operazioni:

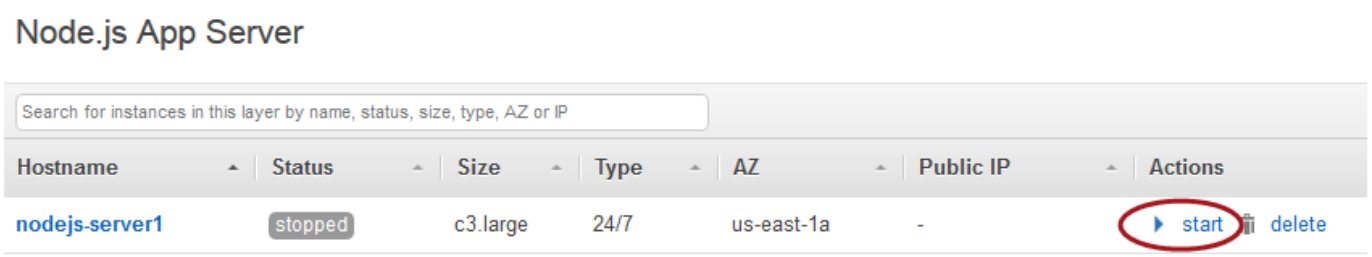
   - Nel riquadro di navigazione del servizio, scegliere Instances (Istanze):



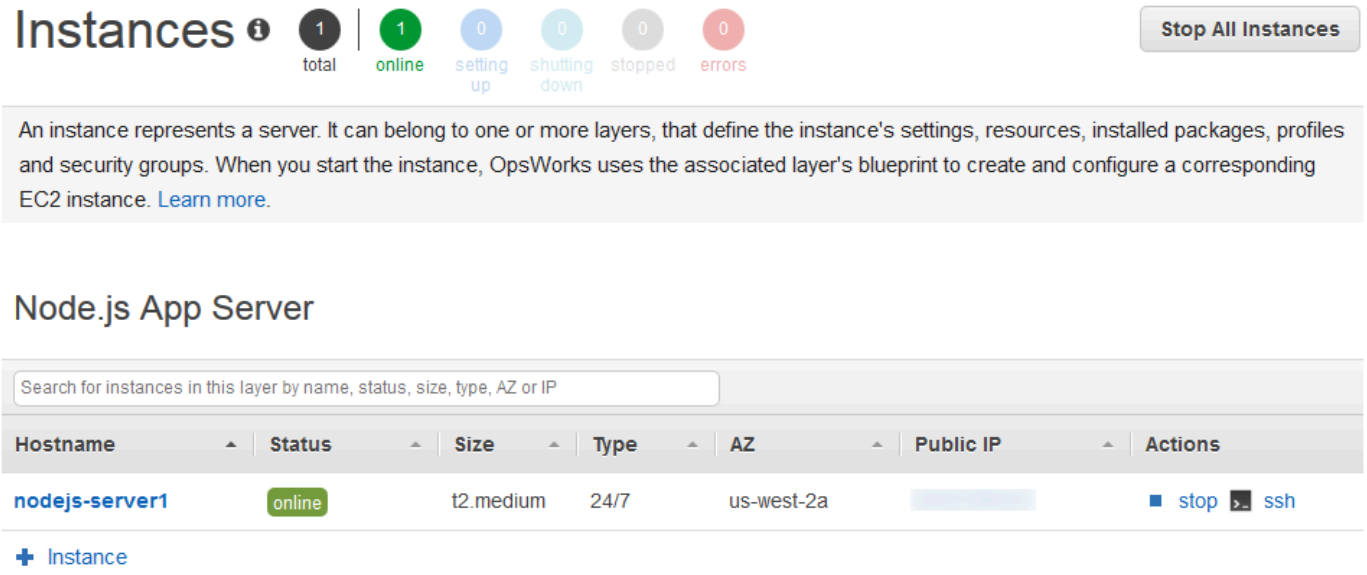
- Nella pagina My Sample Stack (Linux) (Stack di esempio personale (Linux)), scegliere Instances (Istanze):



2. Nella pagina Instances (Istanze), per Node.js App Server (Server app Node.js), nodejs-server1, scegliere start (avvio):



3. Non procedere finché la voce online non diventa verde brillante. Se viene visualizzato un messaggio di errore, consulta la Guida al debugging e alla risoluzione dei problemi.)
4. Durante la configurazione dell'istanza, AWS OpsWorks Stacks distribuisce l'app sull'istanza.
5. I risultati devono essere simili a quelli riportati nel seguente screenshot prima di continuare (se viene visualizzato un messaggio di errore, consulta la Guida al debugging e alla risoluzione dei problemi.):



Ora disponi di un'istanza con un'app distribuita sull'istanza.

Nella fase successiva procederai al test dell'app sull'istanza.

Fase 4. Test dell'app distribuita sull'istanza

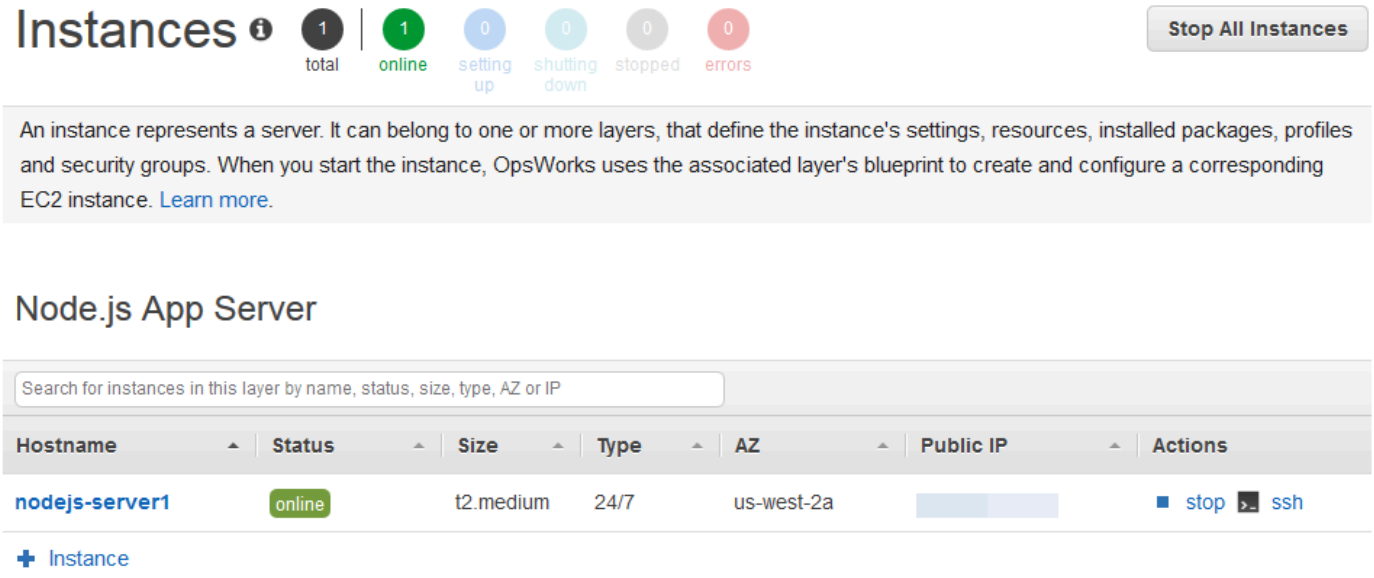
> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

A questo punto puoi eseguire il test della distribuzione dell'app sull'istanza.

Per testare la distribuzione sull'istanza

1. Con la pagina Instances (Istanze) visualizzata nella fase precedente, per Node.js App Server (Server app Node.js), nodejs-server1, Public IP (IP pubblico), scegliere l'indirizzo IP.



2. Nella pagina Web di presentazione, nella casella di testo Leave a comment (Lascia un commento) immettere un commento, quindi scegliere Send (Invia) per testare l'app. L'app aggiunge il commento alla pagina Web. Continuare a immettere commenti e scegliere Send (Invia).



3. Se hai un account Twitter, scegli Tweet o Segui @ AWSOps Works e segui le istruzioni sullo schermo per twittare sull'app o seguire @ Works. AWSOps

A questo punto hai testato correttamente l'app distribuita sull'istanza.

Nei passaggi rimanenti, puoi utilizzare la console AWS OpsWorks Stacks per esplorare le impostazioni dello stack e dei suoi componenti. Nella fase successiva, puoi iniziare l'esplorazione esaminando le impostazioni dello stack.

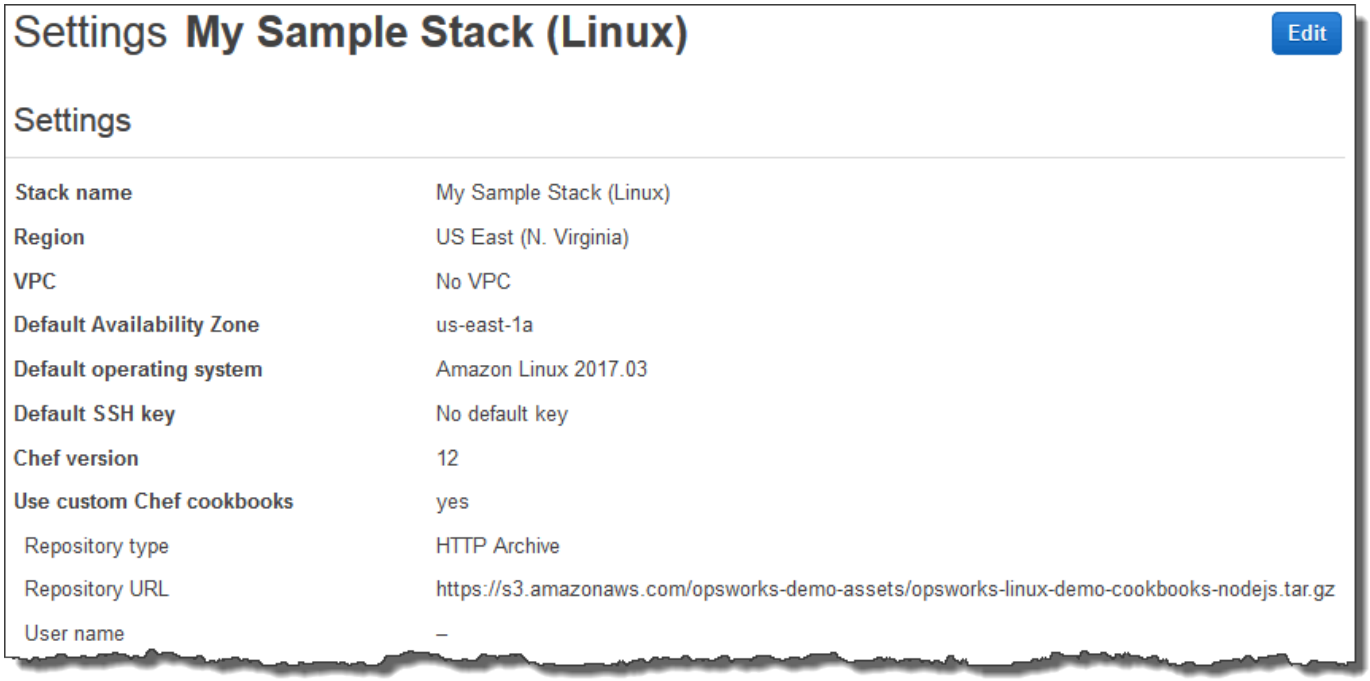
Fase 5. Esplorazione delle impostazioni dello stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Esamina come AWS OpsWorks Stacks ha impostato lo stack.

Per visualizzare le impostazioni dello stack

1. Nel riquadro di navigazione del servizio, scegliere Stack (Stack). Viene visualizzata la pagina My Sample Stack (Linux) (Stack di esempio personale (Linux)).
2. Scegliere Stack Settings (Impostazioni stack). Viene visualizzata la pagina Settings My Sample Stack (Impostazioni stack di esempio personale):



Per ulteriori informazioni sulle impostazioni, scegli Edit (Modifica), quindi passa il puntatore del mouse sopra ogni impostazione. Non tutte le impostazioni sono associate a descrizioni a video. Per ulteriori informazioni su queste impostazioni, consultare Creare un nuovo stack.

Per esplorare il ricettario Chef utilizzato in questa procedura dettagliata, apri il repository opsworks-linux-demo-cookbooks-nodejs su. GitHub

Nella fase successiva, potrai esplorare le impostazioni del livello.

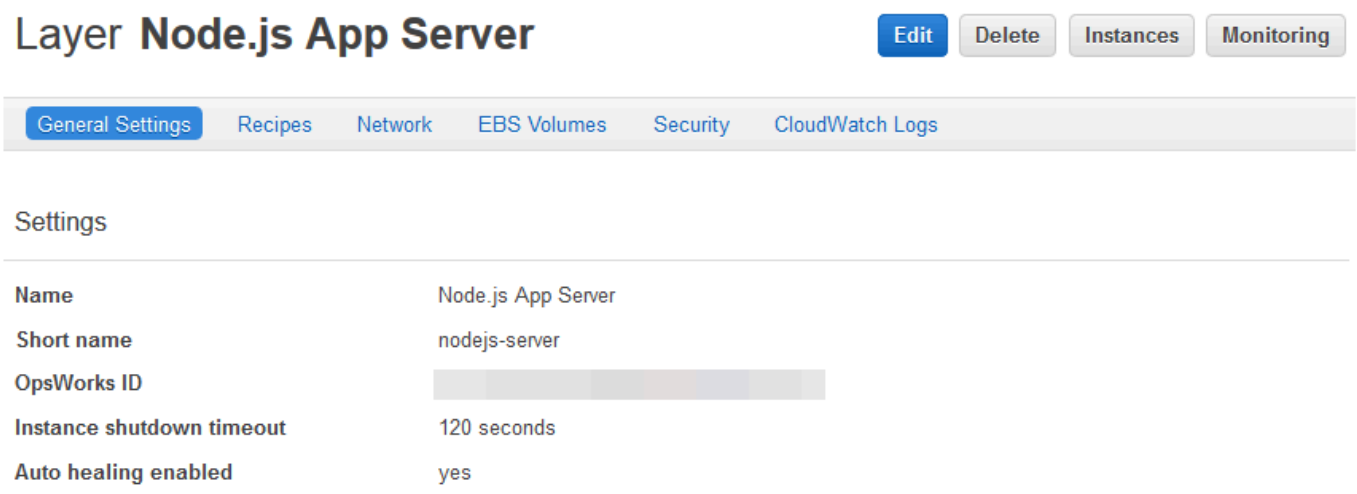
Fase 6. Esplorazione delle impostazioni del livello

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Esamina come AWS OpsWorks Stacks ha impostato il livello.

Per visualizzare le impostazioni del livello

1. Nel riquadro di navigazione del servizio, scegliere Layers (Livelli). Viene visualizzata la pagina Layers (Livelli).
2. Scegliere Node.js App Server (Server app Node.js). Viene visualizzata la pagina Layer Node.js App Server (Server app Node.js livello). Per visualizzare le impostazioni del livello, scegliere General Settings (Impostazioni generali), Recipes (Ricette), Network (Rete), EBS Volumes (Volumi EBS) e Security (Sicurezza):



Per ulteriori informazioni sulle impostazioni, scegli Edit (Modifica), quindi passa il puntatore del mouse sopra ogni impostazione. Non tutte le impostazioni sono associate a descrizioni a video. Per ulteriori informazioni su queste impostazioni, consultare Modifica della configurazione di un livello OpsWorks .

Nella fase successiva, potrai esplorare le impostazioni dell'istanza e i log.

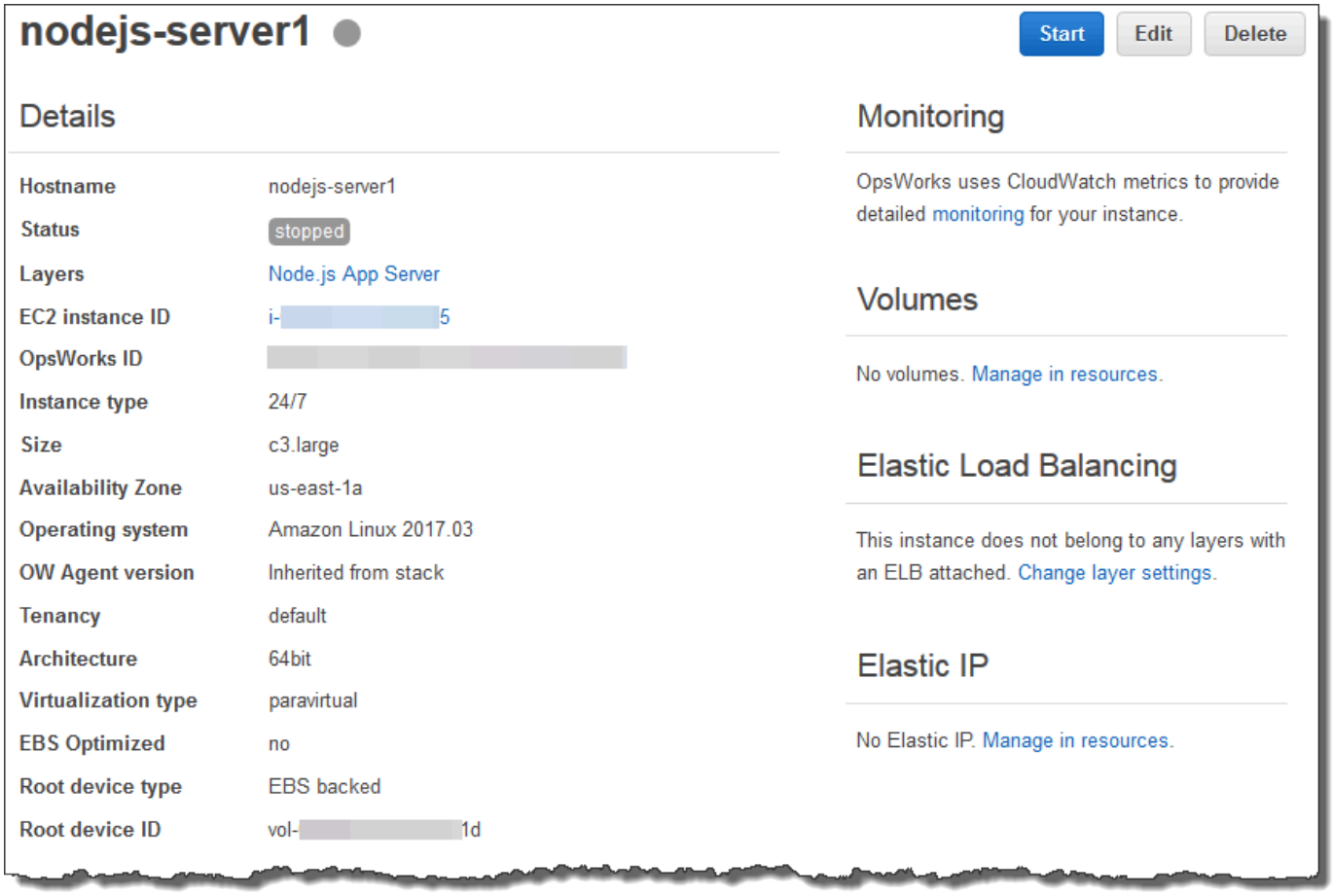
Fase 7. Esplorazione di impostazioni e log dell'istanza

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

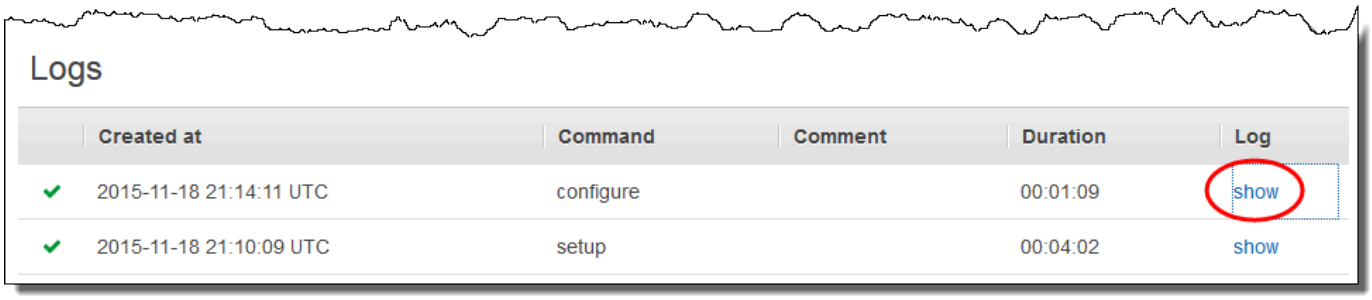
Esamina le impostazioni utilizzate da AWS OpsWorks Stacks per avviare l'istanza. Puoi anche esaminare i log delle istanze creati da AWS OpsWorks Stacks.

Per visualizzare le impostazioni e i log dell'istanza

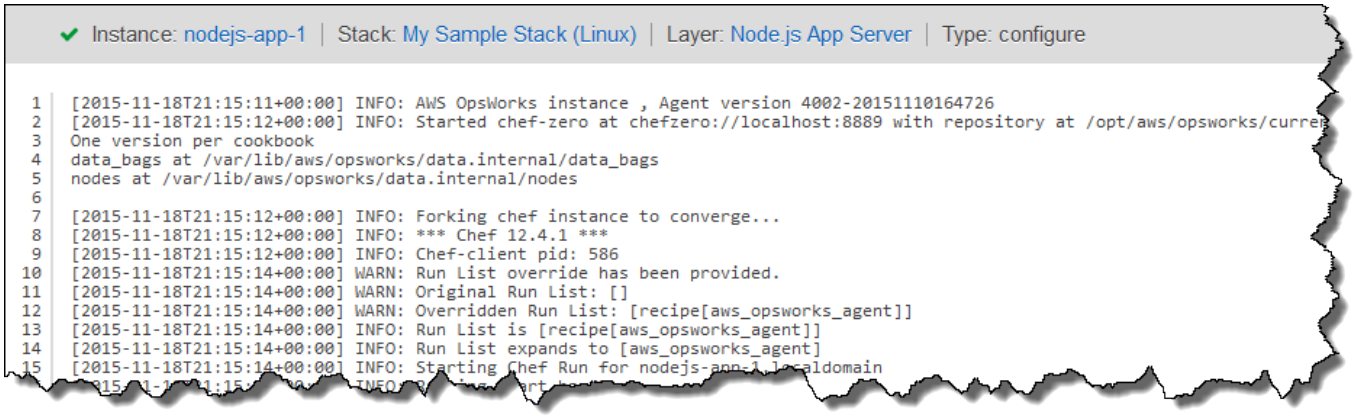
1. Nel riquadro di navigazione del servizio, scegliere Instances (Istanze). Viene visualizzata la pagina Instances (Istanze).
2. In Node.js App Server (Server app Node.js), scegliere nodejs-server1. Viene visualizzata la pagina delle proprietà dell'istanza.



3. Per esplorare i log dell'istanza, nella Logs (Log), per Log, scegliere show (mostra).



4. AWS OpsWorks Stacks visualizza il registro in una scheda separata del browser Web.



Per ulteriori informazioni sul significato di alcune impostazioni dell'istanza, torna alla pagina nodejs-server1, scegli Stop (Arresta) e quando viene visualizzato il messaggio di conferma scegli Stop (Arresta). Scegli Edit (Modifica) dopo che nel campo Status (Stato) la voce stopping (arresto in corso) diventa stopped (arrestato), quindi passa il puntatore del mouse su ciascuna delle impostazioni. Non tutte le impostazioni sono associate a descrizioni a video. Per ulteriori informazioni su queste impostazioni, consultare Aggiunta di un'istanza a un livello.

Dopo aver esaminato le impostazioni, scegli Start (Avvia) per riavviare l'istanza e attendi finché nel campo Status (Stato) non viene visualizzato online. In caso contrario, non sarai in grado di testare l'app in un secondo momento perché l'istanza rimarrà arrestata.

> **Note**
>
> Se desideri accedere all'istanza per esplorarla ulteriormente, devi prima fornire a AWS OpsWorks Stacks le informazioni sulla tua chiave SSH pubblica (che puoi creare con strumenti come ssh-keygen o PuTTYgen), quindi devi impostare le autorizzazioni sullo stack

My Sample Stack (Linux) per consentire all'utente di accedere all'istanza. Per istruzioni, consulta Registrazione della chiave SSH pubblica di un utente e Accesso con SSH.

Nella fase successiva, potrai esplorare le impostazioni dell'app.

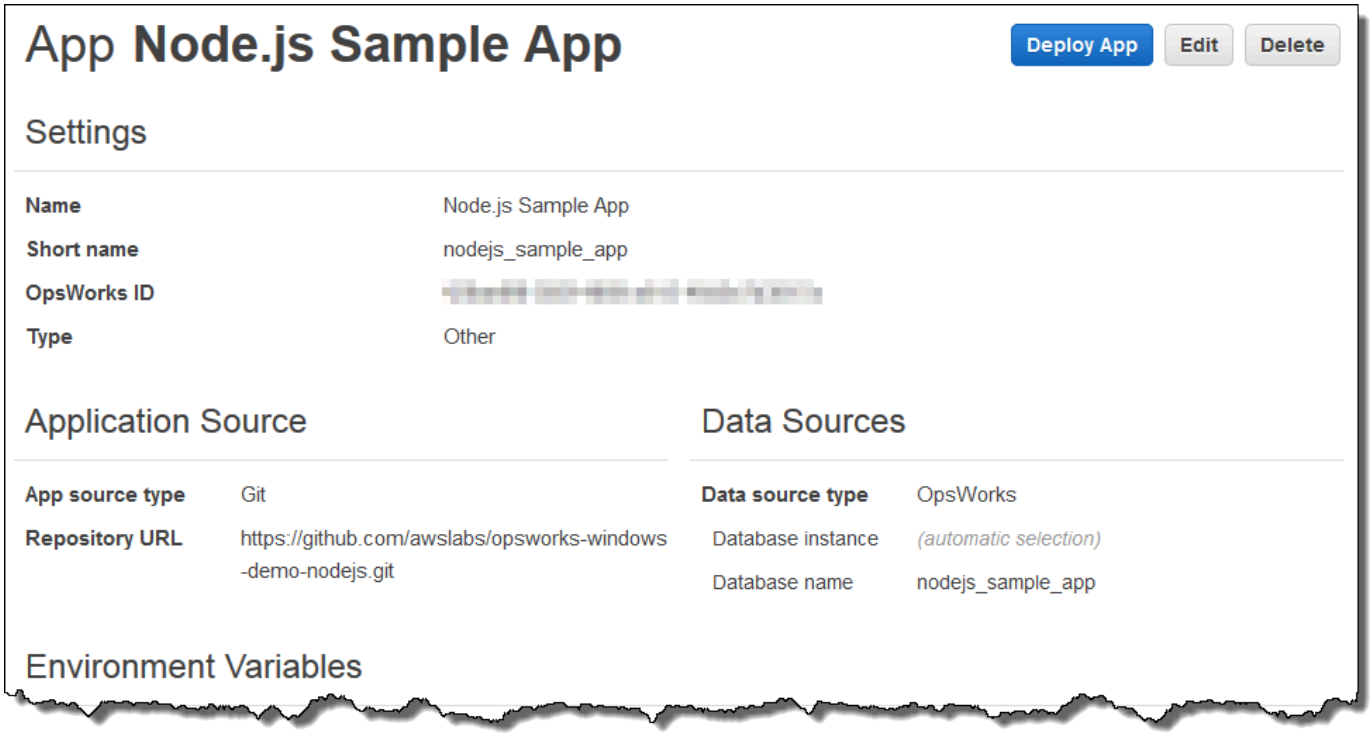
Fase 8. Esplorazione delle impostazioni dell'app

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Esamina le impostazioni utilizzate da AWS OpsWorks Stacks per l'app.

Per visualizzare le impostazioni dell'app

1. Nel riquadro di navigazione del servizio, scegliere Apps (App). Viene visualizzata la pagina Apps (App).
2. Scegliere Node.js Sample App (App di esempio Node.js). Viene visualizzata la pagina Node.js Sample App (App di esempio Node.js):



Per ulteriori informazioni sul significato di alcune impostazioni, scegli Edit (Modifica), quindi passa il puntatore del mouse sopra ogni impostazione. Non tutte le impostazioni sono associate a descrizioni a video. Per ulteriori informazioni su queste impostazioni, consulta Aggiunta di app.

Nella fase successiva, potrai esplorare i rapporti relativi al monitoraggio del livello.

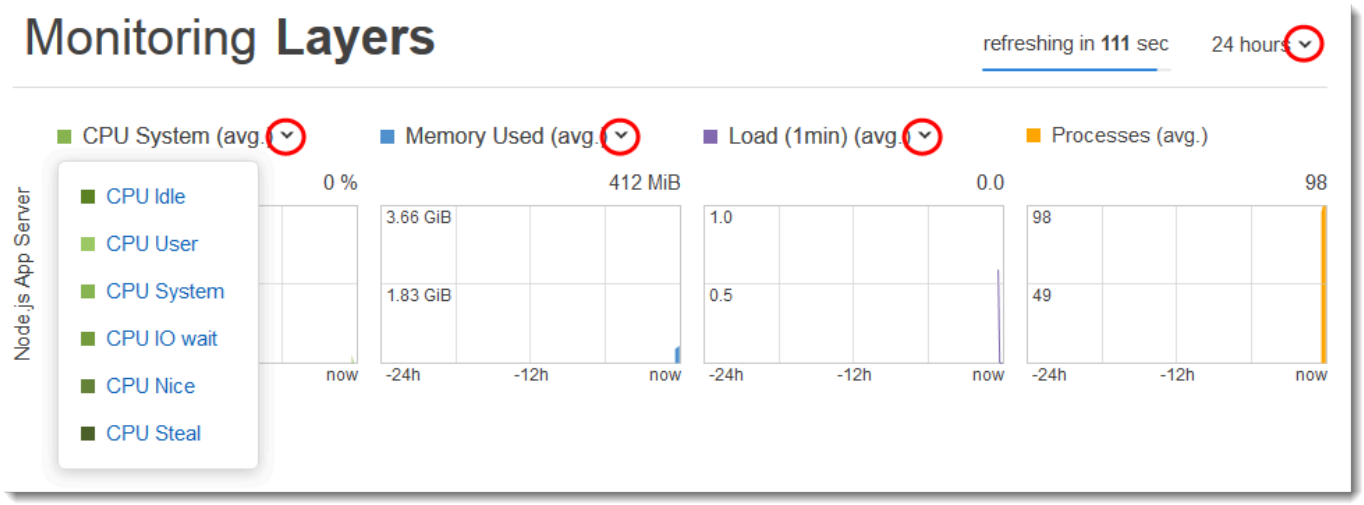
Fase 9. Esplorazione dei report relativi al monitoraggio del livello

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Esamina i report generati da AWS OpsWorks Stacks sulle prestazioni di calcolo del layer.

Per visualizzare i report relativi al monitoraggio del livello

1. Nel riquadro di navigazione dei servizi, scegliere Monitoring (Monitoraggio). Viene visualizzata la pagina Monitoring Layers (Monitoraggio livelli).
2. Per esplorare visualizzazioni aggiuntive, scegliere la freccia accanto a CPU, Memory (Memoria), Load (Carico) e all'ora:



Per informazioni su questi e altri rapporti, consulta Usare Amazon CloudWatch e Monitoraggio.

Nella fase successiva, potrai esplorare altre impostazioni dello stack.

Fase 10. Esplorazione di altre impostazioni dello stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

In questa fase, potrai esaminare le altre impostazioni dello stack.

AWS OpsWorks Stacks non ha eseguito distribuzioni separate, non ha fornito risorse aggiuntive e non ha modificato autorizzazioni aggiuntive come parte di questo stack, quindi non c'è molto interesse nelle pagine Distribuzioni e Comandi, Risorse e Autorizzazioni. Se tuttavia desideri

visualizzare queste impostazioni, scegli rispettivamente Deployments (Distribuzioni), Resources (Risorse) e Permissions (Autorizzazioni) nel riquadro di navigazione del servizio. Se desideri ulteriori informazioni sul significato di queste pagine, consulta Distribuzione di app, Gestione delle risorse e Gestione delle autorizzazioni utente.

Nel passaggio successivo, puoi ripulire le risorse utilizzate per questa procedura dettagliata. AWS Questa fase è opzionale. Potresti voler continuare a utilizzare queste AWS risorse mentre continui a saperne di più su AWS OpsWorks Stacks. Tuttavia, il mantenimento di queste AWS risorse potrebbe comportare alcuni addebiti continui sul tuo AWS account. Se desideri conservare queste AWS risorse per un uso successivo, ora hai completato questa procedura dettagliata e puoi passare a. Fasi successive

Fase 11. Pulizia (opzionale)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per evitare addebiti aggiuntivi AWS sul tuo account, puoi eliminare l'app e AWS le risorse utilizzate per questa procedura dettagliata, inclusi l'istanza e lo stack Stacks. AWS OpsWorks (Per ulteriori informazioni, consulta la sezione Prezzi).AWS OpsWorks Tuttavia, potresti voler continuare a utilizzare queste AWS risorse man mano che continui a saperne di più su AWS OpsWorks Stacks. Se desideri mantenere disponibili queste AWS risorse, ora hai completato questa procedura dettagliata e puoi passare a. Fasi successive

I contenuti archiviati nelle risorse create per questa procedura guidata possono contenere informazioni identificative personali. Se non si desidera che queste informazioni vengano archiviate da AWS, seguire le fasi in questo argomento.

Per eliminare l'app dallo stack

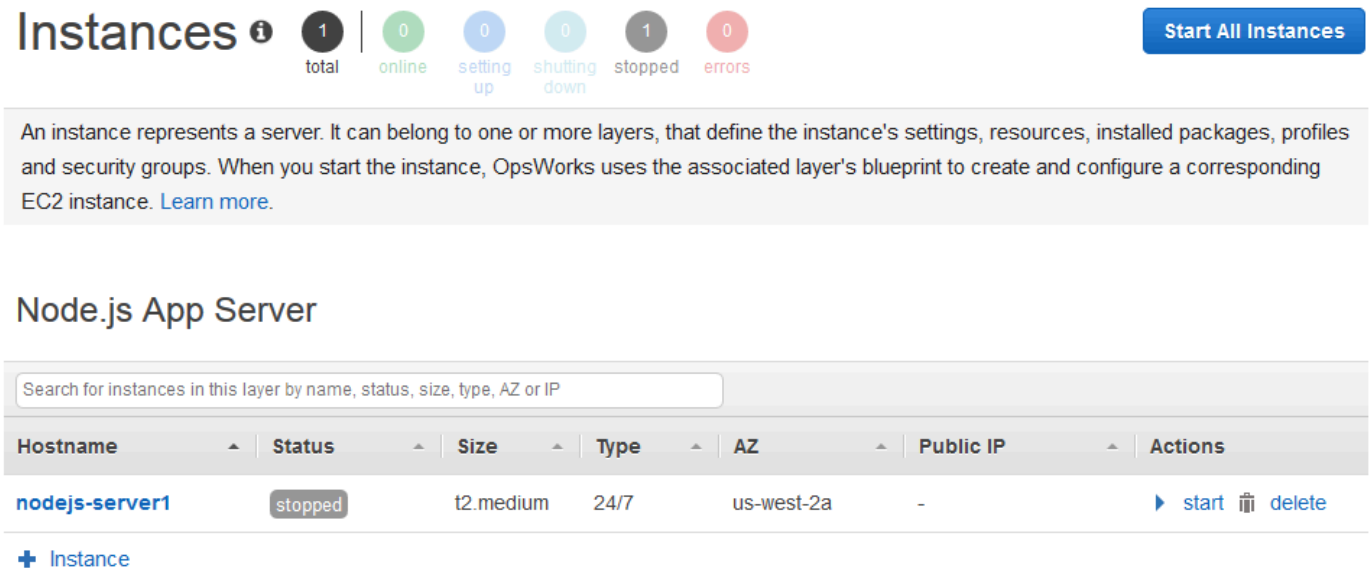
1. Nel riquadro di navigazione del servizio, scegliere Apps (App). Viene visualizzata la pagina Apps (App).

2. In Node.js Sample App (App di esempio Node.js), per Actions (Operazioni), scegliere delete (eliminazione). Quando viene visualizzato il messaggio di conferma, scegliere Delete (Elimina). Dopo aver eliminato l'app, viene visualizzato il messaggio No apps (Nessuna applicazione).

Per eliminare l'istanza per lo stack

1. Nel riquadro di navigazione del servizio, scegliere Instances (Istanze). Viene visualizzata la pagina Instances (Istanze).
2. In Node.js App Server (Server app Node.js), per nodejs-server1, in Actions (Operazioni), scegliere stop (arresto). Quando viene visualizzato il messaggio di conferma, scegliere Stop (Arresta).

   Questo processo può richiedere alcuni minuti. Al termine AWS OpsWorks di Stacks, vengono visualizzati i seguenti risultati.



3. In Actions (Operazioni), scegliere delete (eliminazione). Quando viene visualizzato il messaggio di conferma, scegliere Delete (Elimina). L'istanza viene eliminata e viene visualizzato il messaggio No instances (Nessuna istanza).

Per eliminare lo stack

1. Nel riquadro di navigazione del servizio, scegliere Stack (Stack). Viene visualizzata la pagina My Sample Stack (Linux) (Stack di esempio personale (Linux)).

2. Scegliere Delete Stack (Elimina stack). Quando viene visualizzato il messaggio di conferma, scegliere Delete (Elimina). Lo stack viene eliminato e viene visualizzata la pagina OpsWorksDashboard.

Facoltativamente, puoi eliminare l'utente e la coppia di EC2 chiavi Amazon che hai usato per questa procedura dettagliata, se non desideri riutilizzarli per accedere ad altri AWS servizi e istanze. EC2 Per istruzioni, consulta Eliminazione di un utente IAM e di coppie di EC2 chiavi Amazon e istanze Linux.

A questo punto hai completato la procedura guidata. Per ulteriori informazioni, consulta Fasi successive.

Fasi successive

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ora che hai completato questa procedura dettagliata, puoi saperne di più sull'uso di Stacks: AWS OpsWorks

- Esercitati a ricreare manualmente questa pila da solo usando Stacks. AWS OpsWorks Consultare Nozioni di base: Linux.
- Esplora il ricettario e l'app che AWS OpsWorks Stacks ha usato per questa procedura dettagliata. Consulta Approfondimento: esplorazione del libro di ricette utilizzato in questa procedura guidata e Approfondimento: esplorazione dell'app utilizzata in questa procedura guidata nella procedura guidata di integrazione Nozioni di base: Linux.
- Esercitati a usare AWS OpsWorks Stacks con istanze Windows. Consultare Nozioni di base: Windows.
- Scopri altre informazioni sugli stack eseguendo l'operazione Creare un nuovo stack.
- Scopri altre informazioni sui livelli eseguendo l'operazione Modifica della configurazione di un livello OpsWorks .

- Scopri altre informazioni sulle istanze eseguendo l'operazione Aggiunta di un'istanza a un livello.
- Scopri altre informazioni sulle app eseguendo l'operazione Distribuzione di app.
- Ulteriori informazioni su Libri di ricette e ricette.
- Crea il tuo libro di ricette. Consultare Nozioni di base: libri di ricette.
- Scopri altre informazioni sul controllo dell'accesso agli stack consultando l'argomento Sicurezza e autorizzazioni.

Nozioni di base sugli stack Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

In questa procedura dettagliata, imparerai come utilizzare AWS OpsWorks Stacks per creare un ambiente applicativo Node.js. Al termine, avrai un'istanza Amazon Elastic Compute Cloud (Amazon EC2) che esegue Chef 12, un server HTTP Node.js e un'app Web che puoi utilizzare per interagire con Twitter e lasciare commenti su una pagina Web.

Chef è un framework di terze parti per la configurazione e la manutenzione dei server, come EC2 le istanze, e il modo in cui le app vengono distribuite e gestite su tali server. Se non conosci Chef, dopo aver completato questa procedura dettagliata, ti consigliamo di saperne di più su Chef in modo da poter sfruttare appieno tutto ciò che Stacks ha AWS OpsWorks da offrire. Per ulteriori informazioni, visita il sito Web Learn Chef.

AWS OpsWorks Stacks supporta quattro distribuzioni Linux: Amazon Linux, Ubuntu Server, CentOS e Red Hat Enterprise Linux. Per questa procedura dettagliata, utilizziamo Ubuntu Server. AWS OpsWorks Stacks funziona anche con Windows Server. Sebbene sia disponibile una procedura dettagliata equivalente per gli stack di Windows Server, ti consigliamo di completare prima questa procedura dettagliata per apprendere concetti di base su AWS OpsWorks Stacks e Chef che non vengono ripetuti qui. Dopo aver completato questa procedura guidata, esegui la procedura guidata Nozioni di base: Windows.

Argomenti

- Fase 1: completamento dei prerequisiti
- Fase 2. Creazione di uno stack
- Fase 3. Aggiunta di un livello allo stack
- Fase 4. Specifica dell'app da distribuire sull'istanza
- Fase 5. Avvio di un'istanza
- Fase 6. Distribuzione dell'app sull'istanza
- Fase 7. Test dell'app distribuita sull'istanza
- Fase 8. Pulizia (opzionale)
- Fasi successive
- Approfondimento: esplorazione del libro di ricette utilizzato in questa procedura guidata
- Approfondimento: esplorazione dell'app utilizzata in questa procedura guidata

Fase 1: completamento dei prerequisiti

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Prima di poter avviare la procedura guidata, completa le seguenti fasi di configurazione. Questi passaggi di configurazione includono la registrazione di un AWS account, la creazione di un utente amministrativo e l'assegnazione delle autorizzazioni di accesso a Stacks. AWS OpsWorks

Se hai già completato la procedura guidata Nozioni di base: esempio, hai già completato i prerequisiti per questa procedura guidata, pertanto puoi passare direttamente a Fase 2. Creazione di uno stack.

Argomenti

- Registrati per un Account AWS

- Crea un utente con accesso amministrativo
- Assegna le autorizzazioni di accesso al servizio

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Parte della procedura di registrazione prevede la ricezione di una telefonata o di un messaggio di testo e l'immissione di un codice di verifica sulla tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Assegna le autorizzazioni di accesso al servizio

Abilita l'accesso al servizio AWS OpsWorks Stacks (e ai servizi correlati su cui si basa AWS OpsWorks Stacks) aggiungendo le `AmazonS3FullAccess` autorizzazioni `AWSOpsWorks_FullAccess` e al tuo ruolo o utente.

Per ulteriori informazioni sull'aggiunta di autorizzazioni, consulta Aggiungere i permessi di identità IAM (console).

A questo punto hai completato tutte le fasi di configurazione. Ora puoi avviare questa procedura guidata.

Fase 2. Creazione di uno stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizzerai la console AWS OpsWorks Stacks per creare uno stack. Uno stack è una raccolta di istanze e AWS risorse correlate che hanno uno scopo comune e che desiderate gestire insieme. Per ulteriori informazioni, consulta Stack. Per questa procedura guidata è disponibile solo un'istanza.

Prima di iniziare, completa i prerequisiti, se non hai già eseguito questa operazione.

Per creare lo stack

1. Accedi a AWS Management Console e apri la AWS OpsWorks console all'indirizzo. https://console.aws.amazon.com/opsworks/
2. Eseguire una delle seguenti operazioni, se possibile:
   - Se viene visualizzata la pagina Welcome to AWS OpsWorks Stacks, scegli Aggiungi il tuo primo stack o Aggiungi il tuo primo AWS OpsWorks stack Stacks (entrambe le scelte hanno la stessa funzione). Viene visualizzata la pagina Add stack (Aggiungi stack).
   - Se viene visualizzata la pagina OpsWorks Dashboard, scegli Aggiungi stack. Viene visualizzata la pagina Add stack (Aggiungi stack).

3. Con la pagina Add stack (Aggiungi stack) visualizzata, scegliere Chef 12 stack (Stack Chef 12), se l'opzione non è già stata selezionata automaticamente.
4. Nella casella Stack name (Nome stack) digitare il nome dello stack, ad esempio **`MyLinuxDemoStack`**. È possibile utilizzare un nome diverso, ma è necessario assicurarsi di sostituirlo a `MyLinuxDemoStack` durante l'esecuzione di questa procedura guidata.
5. Per Regione, scegli Stati Uniti occidentali (Oregon).
6. In VPC, eseguire una delle seguenti operazioni:
   - Se è disponibile un VPC, sceglierlo. Per ulteriori informazioni, consulta Esecuzione di uno stack in un VPC.
   - In caso contrario, scegliere No VPC (Nessun VPC).
7. In Default operating system (Sistema operativo predefinito), scegliere Linux e Ubuntu 18.04 LTS.
8. In Use custom Chef cookbooks (Utilizza i libri di ricette di Chef personalizzati), scegliere Yes (Sì).
9. In Repository type (Tipo di repository), scegliere Http Archive (Archivio Http).
10. Nel campo Repository URL (URL repository), digitare **`https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-cookbooks-nodejs.tar.gz`**
11. Non modificare le impostazioni predefinite per le seguenti opzioni:
    - Default Availability Zone (Zona di disponibilità predefinita) (us-west-2a)
    - Default SSH key (Chiave SSH predefinita) (Do not use a default SSH key (Non utilizzare una chiave SSH predefinita))
    - User name (Nome utente) (vuoto)
    - Password (vuoto)
    - Stack color (Colore stack) (blu scuro)
12. Scegli Advanced (Avanzato).
13. Per il ruolo IAM, esegui una delle seguenti operazioni (per ulteriori informazioni, consultaConsentire a AWS OpsWorks Stacks di agire per tuo conto):
    - Se aws-opsworks-service-roleè disponibile, sceglilo.
    - Se non aws-opsworks-service-roleè disponibile, scegli Nuovo ruolo IAM.
14. Per il profilo predefinito dell'istanza IAM, esegui una delle seguenti operazioni (per ulteriori informazioni, consultaSpecificazione delle autorizzazioni per le app in esecuzione su istanze EC2):

- Se sono disponibili aws-opsworks-ec2 ruoli, sceglili.
- Se aws-opsworks-ecil doppio ruolo non è disponibile, scegli Nuovo profilo di istanza IAM.

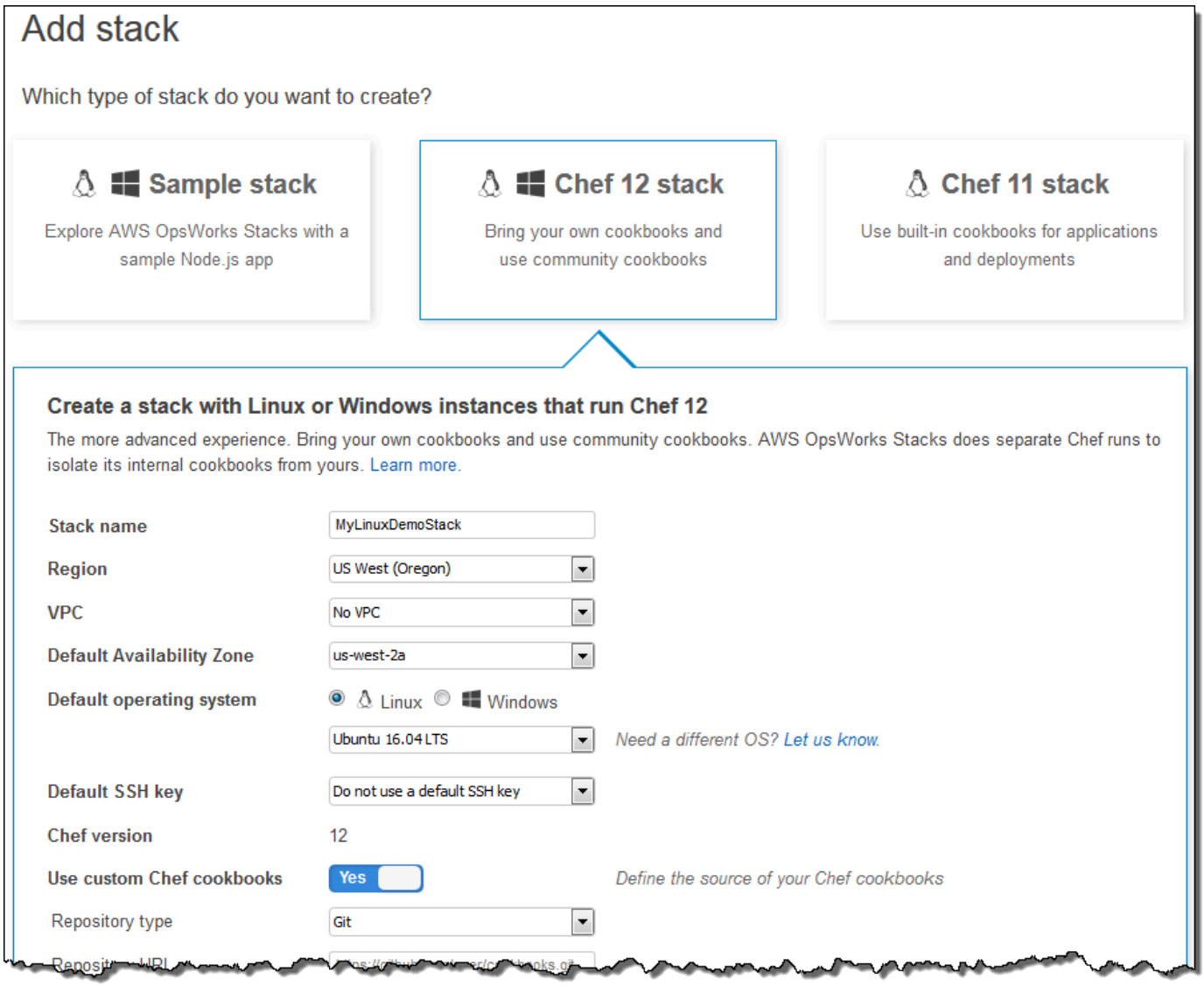
15. In API endpoint region (Regione endpoint API), scegliere l'endpoint API regionale a cui si desidera associare lo stack. Se desideri che lo stack si trovi nella regione Stati Uniti occidentali (Oregon) all'interno dell'endpoint regionale degli Stati Uniti orientali (Virginia settentrionale), scegli us-east-1. Se desideri che lo stack si trovi sia nella regione Stati Uniti occidentali (Oregon) sia associato all'endpoint regionale degli Stati Uniti occidentali (Oregon), scegli us-west-2.

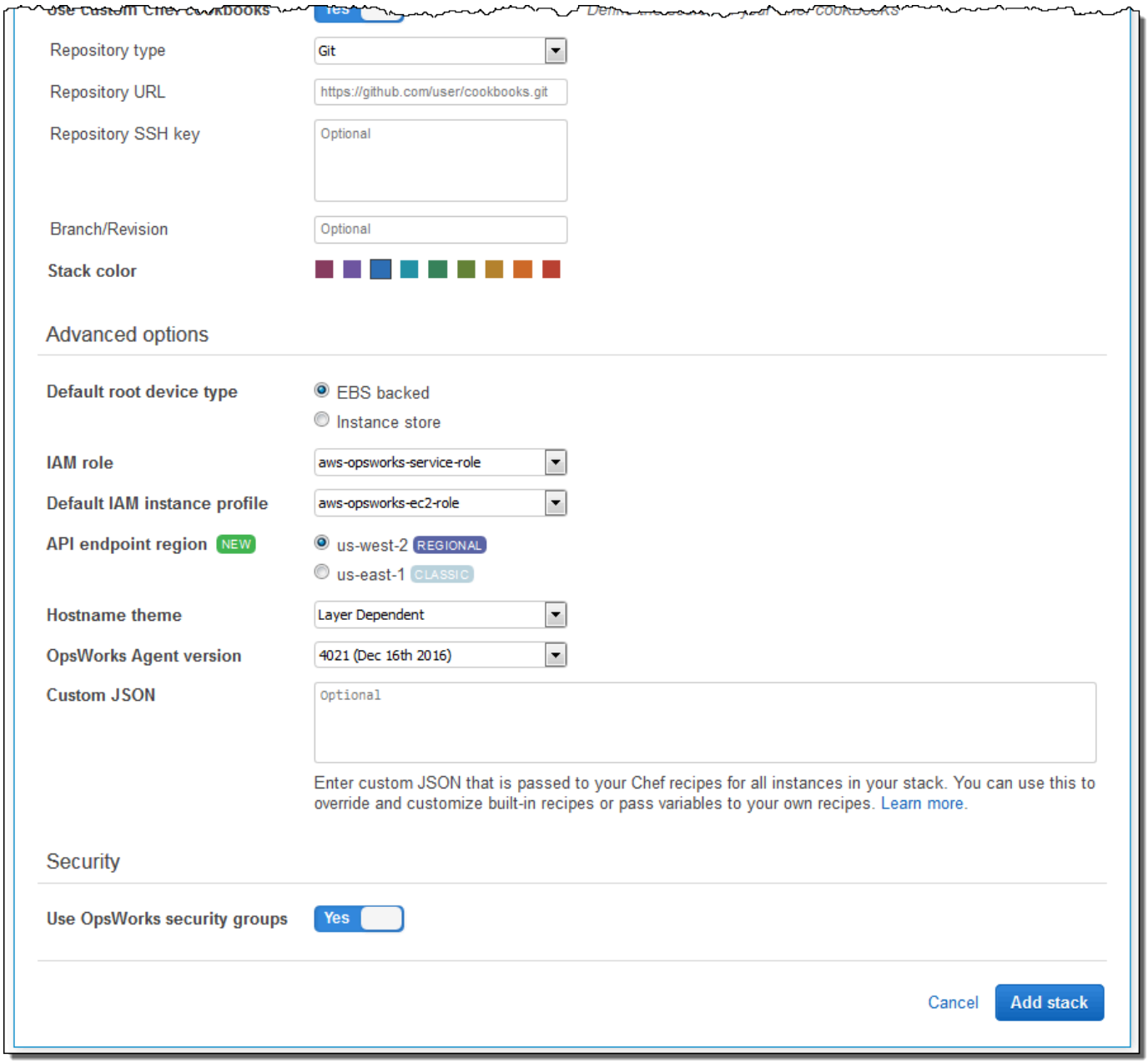
> **Note**
>
> L'endpoint regionale degli Stati Uniti orientali (Virginia settentrionale) include versioni precedenti Regioni AWS per motivi di compatibilità con le versioni precedenti, ma è consigliabile scegliere l'endpoint regionale più vicino a quello in cui gestisci. AWS Per ulteriori informazioni, consulta Supporto nelle regioni.

16. Non modificare le impostazioni predefinite per le seguenti opzioni:

- Default root device type (Tipo di dispositivo radice predefinito) (EBS backed (Supportato da EBS))
- Hostname theme (Tema nome host) (Layer Dependent (Dipendente dal livello))
- OpsWorks Versione dell'agente (versione più recente)
- Custom JSON (JSON personalizzato) (vuoto)
- Usa gruppi OpsWorks di sicurezza (Sì)

17. I risultati dovrebbero corrispondere alle schermate seguenti, ad eccezione forse di VPC, ruolo IAM e profilo di istanza IAM predefinito:


Add stack
Which type of stack do you want to create?
Sample stack
Explore AWS OpsWorks Stacks with a sample Node.js app
Chef 12 stack
Bring your own cookbooks and use community cookbooks
Chef 11 stack
Use built-in cookbooks for applications and deployments
Create a stack with Linux or Windows instances that run Chef 12
The more advanced experience. Bring your own cookbooks and use community cookbooks. AWS OpsWorks Stacks does separate Chef runs to isolate its internal cookbooks from yours. Learn more.
Stack name
MyLinuxDemoStack
Region
US West (Oregon)
VPC
No VPC
Default Availability Zone
us-west-2a
Default operating system
Linux
Windows
Ubuntu 16.04 LTS
Need a different OS? Let us know.
Default SSH key
Do not use a default SSH key
Chef version
12
Use custom Chef cookbooks
Yes
Define the source of your Chef cookbooks
Repository type
Git



18. Scegli Aggiungi stack. AWS OpsWorks Stacks crea lo stack e visualizza la pagina. MyLinuxDemoStack

Ora disponi di uno stack con le impostazioni corrette per questa procedura guidata.

Nella fase successiva, aggiungerai un livello allo stack.

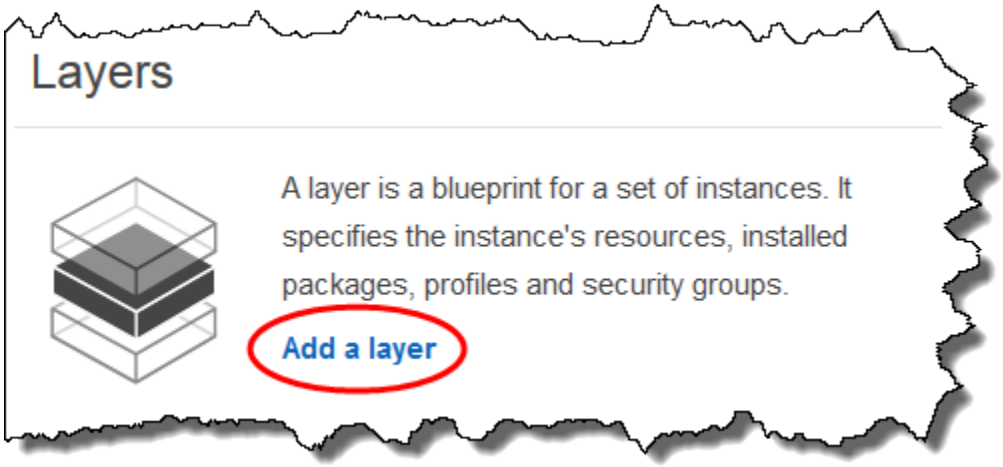
Fase 3. Aggiunta di un livello allo stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

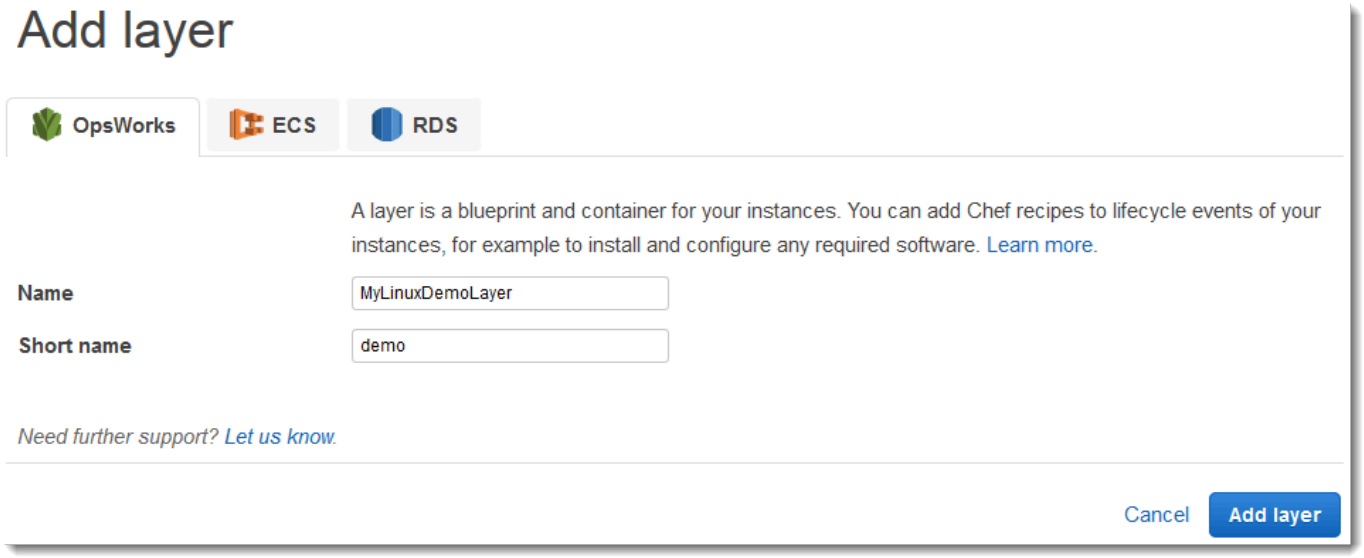
Un layer è un modello per un insieme di istanze, come le istanze Amazon EC2. Specifica informazioni quali, ad esempio, le impostazioni dell'istanza, le risorse, i pacchetti installati e i gruppi di sicurezza. Si aggiunge quindi un livello allo stack. Per ulteriori informazioni sui livelli, consulta Livelli.

Per aggiungere il livello allo stack

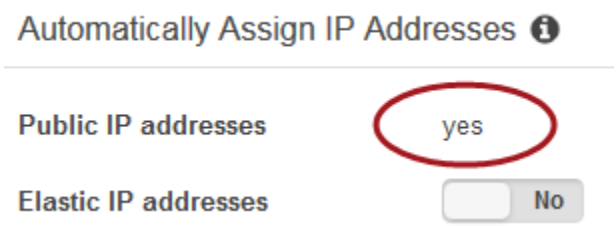
1. Con la MyLinuxDemoStackpagina visualizzata nel passaggio precedente, per Livelli, scegli Aggiungi un livello:



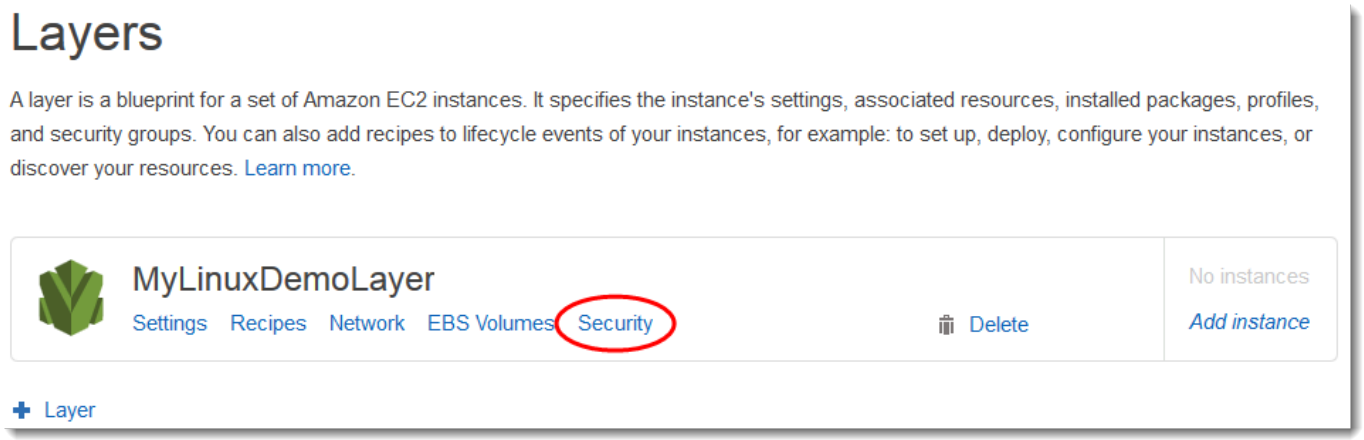
2. Viene visualizzata la pagina Add Layer (Aggiungi livello). Nella OpsWorksscheda, per Nome, digitate**`MyLinuxDemoLayer`**. È possibile utilizzare un nome diverso, ma è necessario assicurarsi di sostituirlo a `MyLinuxDemoLayer` durante l'esecuzione di questa procedura guidata.
3. In Short name (Nome breve), immettere **`demo`**. È anche possibile immettere un valore diverso, assicurandosi però di sostituirlo sempre a `demo` in tutta la procedura guidata:



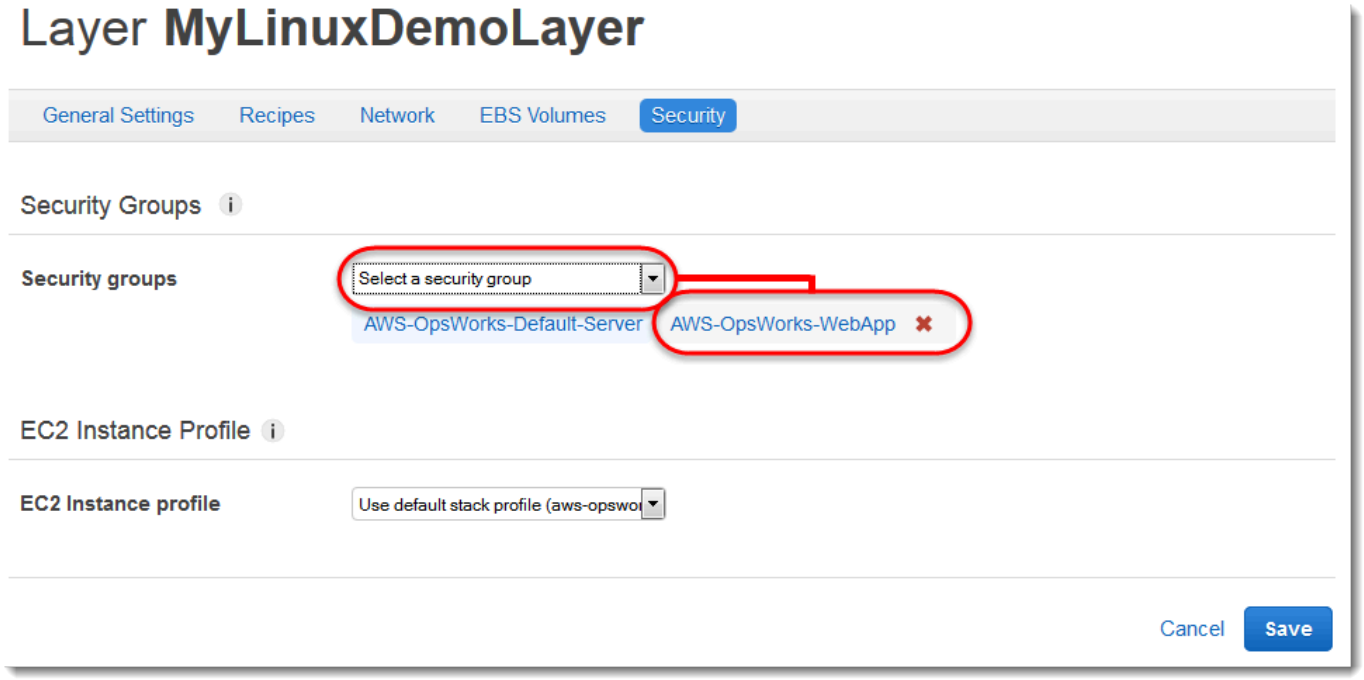
4. Scegli Aggiungi livello. AWS OpsWorks Stacks crea il livello e visualizza la pagina Livelli.
5. Nella pagina Livelli, per MyLinuxDemoLayer, scegli Rete.
6. Nella scheda Network (Rete), in Automatically Assign IP Addresses (Assegna automaticamente indirizzi IP), verificare che l'opzione Public IP addresses (Indirizzi IP pubblici) sia impostata su yes (sì). Se sono state apportate delle modifiche, scegliere Save (Salva).



7. Nella pagina Layers (Livelli), scegliere Security (Sicurezza):



8. La MyLinuxDemoLayer pagina Layer viene visualizzata con la scheda Sicurezza aperta. Per i gruppi di sicurezza, scegli AWS- OpsWorks - WebApp, quindi scegli Salva:



9. Il gruppo di sicurezza `AWS-OpsWorks-WebApp` viene aggiunto al livello. (Questo gruppo di sicurezza consente agli utenti di connettersi all'app sull'istanza più avanti in questa procedura dettagliata. Senza questo gruppo di sicurezza, gli utenti riceveranno un messaggio nel browser Web che indica che non possono connettersi all'istanza.)

Ora disponi di un livello con le impostazioni corrette per questa procedura guidata.

Nella fase successiva specificherai l'app da distribuire sull'istanza.

Fase 4. Specifica dell'app da distribuire sull'istanza

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

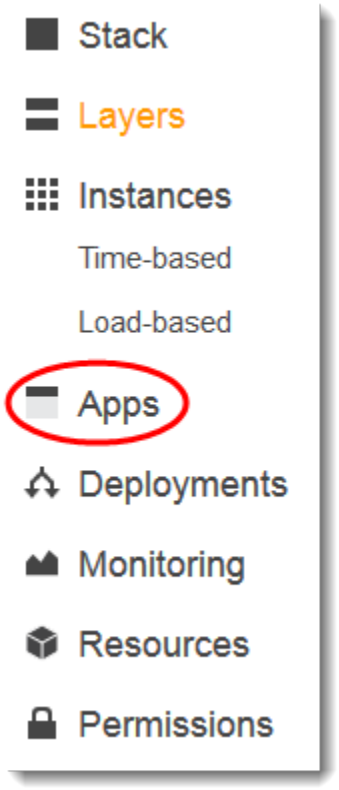
sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Descrivi a AWS OpsWorks Stacks l'app che distribuirai sull'istanza più avanti in questa procedura dettagliata. In questo contesto, AWS OpsWorks Stacks definisce un'app come il codice che desideri eseguire su un'istanza. Per ulteriori informazioni, consulta App.

La procedura in questa sezione è valida per stack Chef 12 e stack di versioni più recenti. Per ulteriori informazioni su come aggiungere app ai livelli negli stack Chef 11, consulta Fase 2.4: Creazione e distribuzione di un'app - Chef 11.

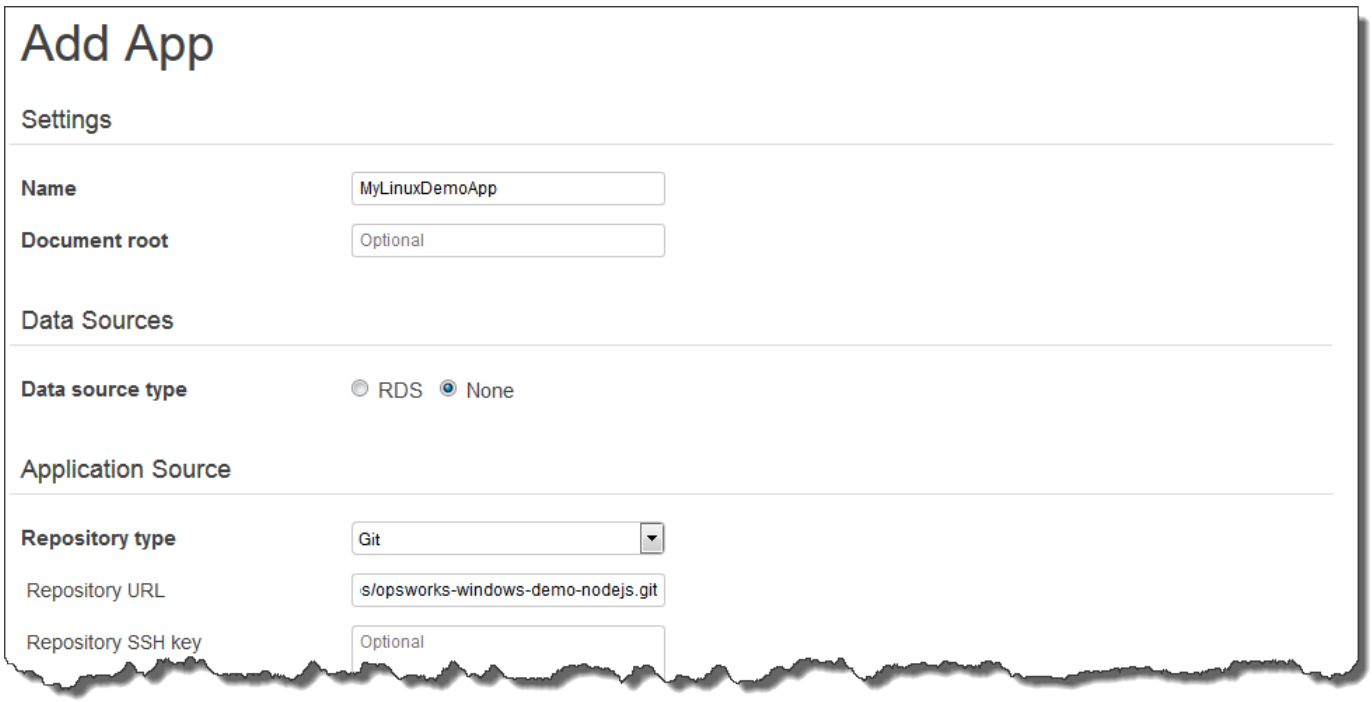
Per specificare l'app da distribuire

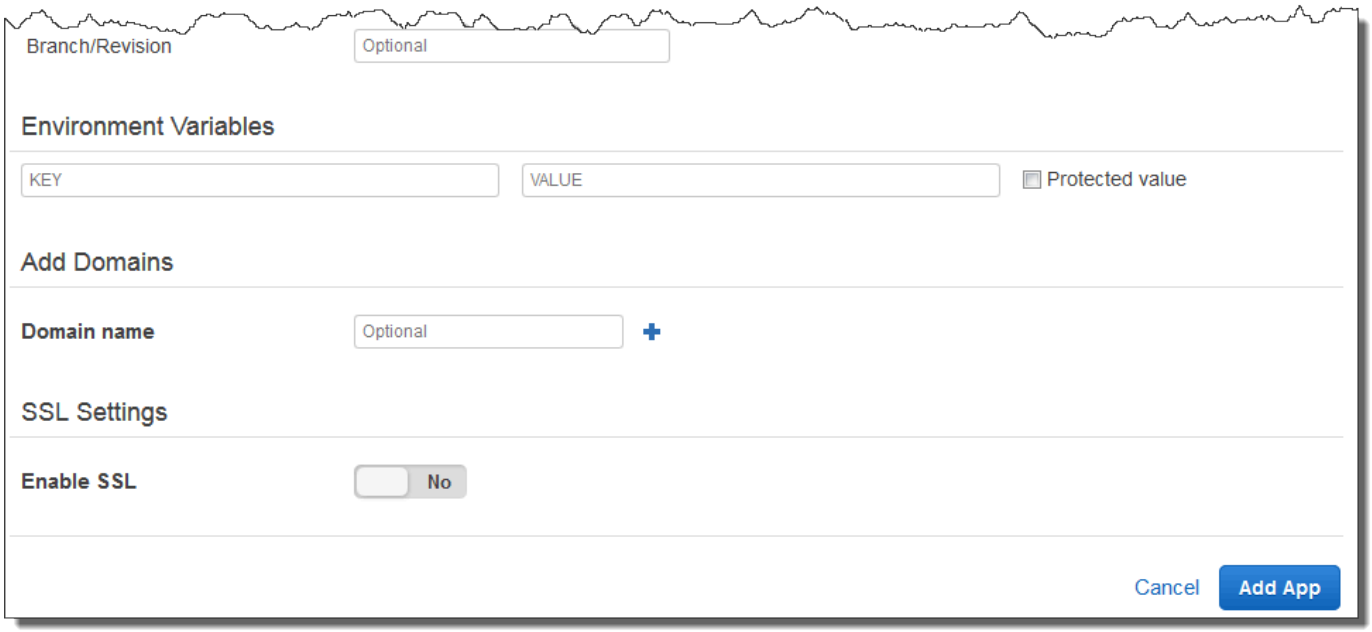
1. Nel riquadro di navigazione del servizio, scegliere Apps (App):



2. Viene visualizzata la pagina Apps (App). Selezionare Add an app (Aggiungi un'app). Viene visualizzata la pagina Add App (Aggiungi app).

3. In Impostazioni, in Nome, digita**`MyLinuxDemoApp`**. È possibile utilizzare un nome diverso, ma è necessario assicurarsi di sostituirlo a `MyLinuxDemoApp` durante l'esecuzione di questa procedura guidata.
4. Per Application Source (Origine dell'applicazione), in Repository URL (URL repository), digitare **`https://github.com/awslabs/opsworks-windows-demo-nodejs.git`**
5. Non modificare le impostazioni predefinite per le seguenti opzioni:
   - Settings (Impostazioni), Document root (Radice documento) (vuoto)
   - Data Sources (Origini dati), Data source type (Tipo di origine dati) (None (Nessuno))
   - Repository type (Tipo di repository) (Git)
   - Repository SSH key (Chiave SSH archivio) (vuoto)
   - Branch/Revision (Ramo/Versione) (vuoto)
   - Environment Variables (Variabili di ambiente) - KEY (CHIAVE) (vuoto), VALUE (VALORE) (vuoto), Protected Value (Valore protetto) deselezionato
   - Add Domains (Aggiungi domini), Domain Name (Nome dominio) (vuoto)
   - SSL Settings (Impostazioni SSL), Enable SSL (Abilita SSL) (No)





6. Scegli Aggiungi app. AWS OpsWorks Stacks aggiunge l'app e visualizza la pagina App.

Ora disponi di un'app con le impostazioni corrette per questa procedura guidata.

Nella fase successiva, avvierai l'istanza.

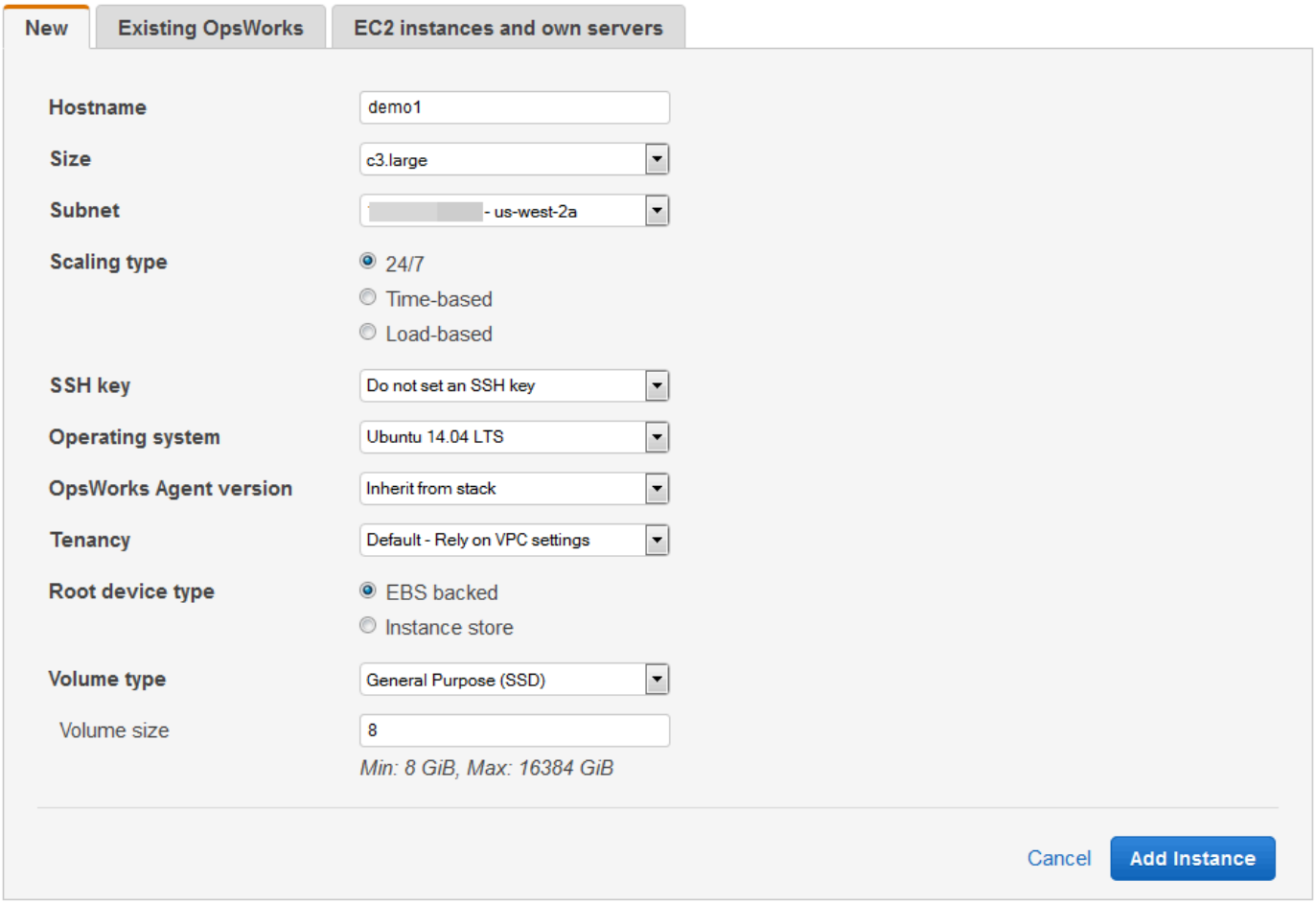
Fase 5. Avvio di un'istanza

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

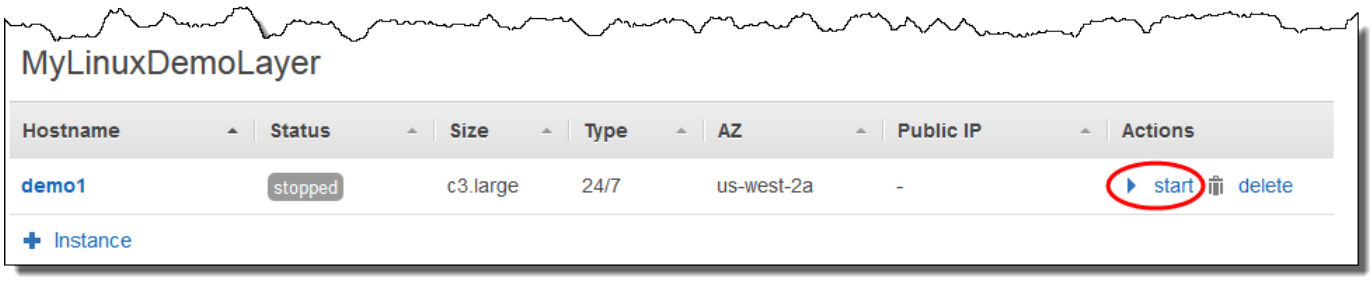
Usa AWS OpsWorks Stacks per avviare un' EC2 istanza Amazon di Ubuntu Server. Questa istanza usa le impostazioni definite nel livello creato in precedenza in questa procedura guidata. Per ulteriori informazioni, consulta Istanze.

Per avviare l'istanza

1. Nel riquadro di navigazione del servizio, scegliere Instances (Istanze). Viene visualizzata la pagina Instances (Istanze).
2. Per MyLinuxDemoLayer, scegli Aggiungi un'istanza.
3. Nella scheda New (Nuova) non modificare le impostazioni predefinite delle seguenti opzioni:
   - Hostname (Nome host) (demo1)
   - Size (Dimensione) (c3.large)
   - Sottorete (*IP address*us-west-2a)
4. Scegli Advanced (Avanzato).
5. Non modificare le impostazioni predefinite per le seguenti opzioni:
   - Scaling type (Tipo di dimensionamento) (24/7)
   - SSH key (Chiave SSH) (Do not use a default SSH key (Non utilizzare una chiave SSH predefinita))
   - Operating system (Sistema operativo) (Ubuntu 18.04 LTS)
   - OpsWorks Versione dell'agente (eredita dallo stack)
   - Tenancy (Default - Rely on VPC settings (Predefinita - Basata su impostazioni VPC))
   - Root device type (Tipo di dispositivo radice) (EBS backed (Supportato da EBS))
   - Volume type (Tipo di volume) (General Purpose (SSD) (Per scopi generici (SSD))
   - Volume size (Dimensioni volume) (8)
6. I risultati saranno simili a quelli illustrati nello screenshot riportato di seguito:



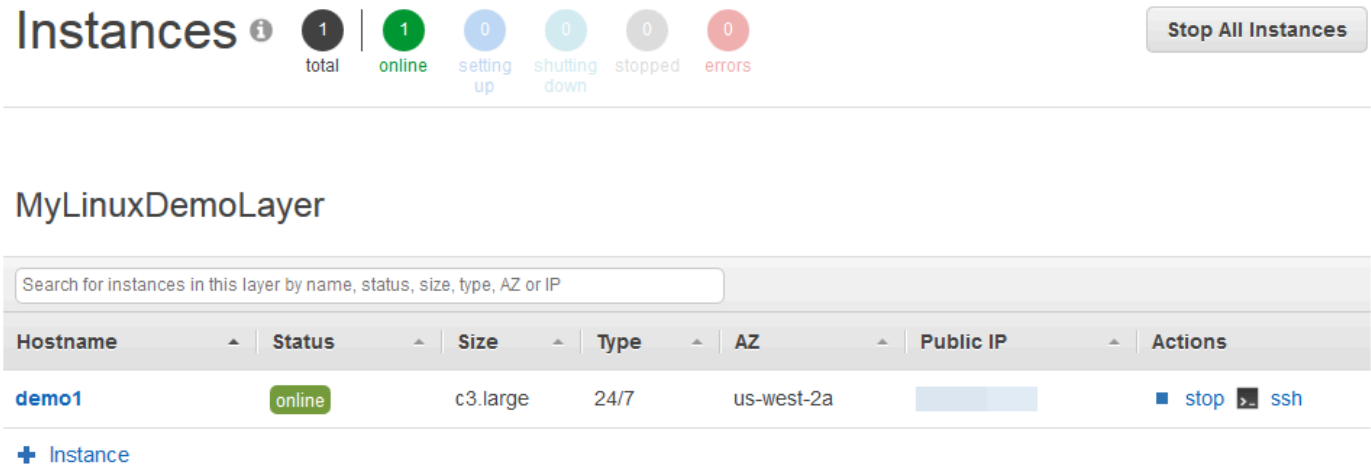
7. Scegli Aggiungi istanza. AWS OpsWorks Stacks aggiunge l'istanza al layer e visualizza la pagina Istanze.
8. Per MyLinuxDemoLayer, per demo1, per Actions, scegliete start:



9. Nei minuti successivi si verificheranno i seguenti eventi:
   - Il valore dell'icona setting up (configurazione in corso) cambierà da 0 a 1.

- L'icona Status (Stato) assumerà in sequenza i seguenti valori: stopped (arrestato), requested (richiesto), pending (in attesa), booting (avvio in corso), running_setup (configurazione in corso) e infine online. Questo processo può richiedere alcuni minuti.
- Quando l'icona Status (Stato) passa al valore online, l'indicatore setting up (configurazione in corso) passa da 1 a 0 e l'icona online passa da 0 a 1 e diventa verde brillante. Non procedere fino a quando l'icona online non diventa verde brillante e mostra il valore 1 (istanza online).

10. I risultati devono corrispondere a quelli riportati nel seguente screenshot prima di continuare (se viene visualizzato un messaggio di errore, consulta la Guida al debugging e alla risoluzione dei problemi):



Ora disponi di un'istanza pronta per la distribuzione dell'app.

> **Note**
>
> Se desideri accedere all'istanza per esplorarla ulteriormente, devi prima fornire a AWS OpsWorks Stacks le informazioni sulla tua chiave SSH pubblica (che puoi creare con strumenti come ssh-keygen o PuTTYgen), quindi devi impostare le autorizzazioni sullo `MyLinuxDemoStack` stack per consentire all'utente di accedere all'istanza. Per istruzioni, consulta Registrazione della chiave SSH pubblica di un utente e Accesso con SSH. Se prevedi di utilizzare SSH per connetterti alle istanze tramite PuTTY, vedi Connessione all'istanza Linux da Windows con PuTTY nella documentazione. AWS

Nella fase successiva distribuirai l'app sull'istanza.

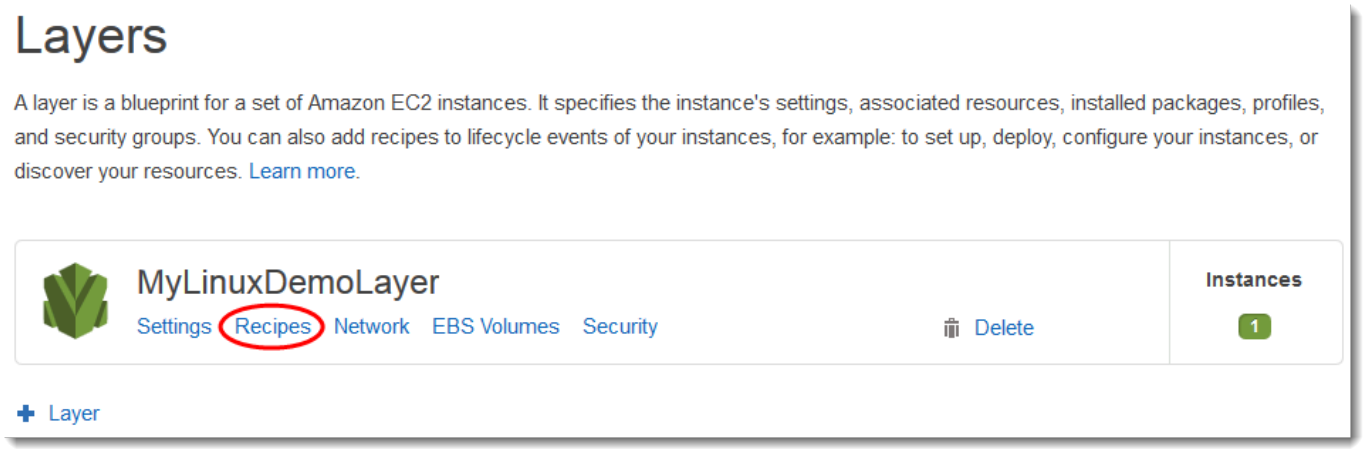
Fase 6. Distribuzione dell'app sull'istanza

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

In questo passaggio, distribuirai l'app dall'istanza GitHub in esecuzione. Per ulteriori informazioni, consulta Distribuzione di app. Prima di distribuire l'app, devi specificare la ricetta da usare per coordinare la distribuzione. Un ricetta è un concetto Chef. Le ricette sono istruzioni, scritte con la sintassi del linguaggio Ruby, che specificano le risorse da utilizzare e l'ordine in cui tali risorse vengono applicate. Per ulteriori informazioni, consulta la sezione relativa all'introduzione delle ricette nel sito Web Learn Chef.

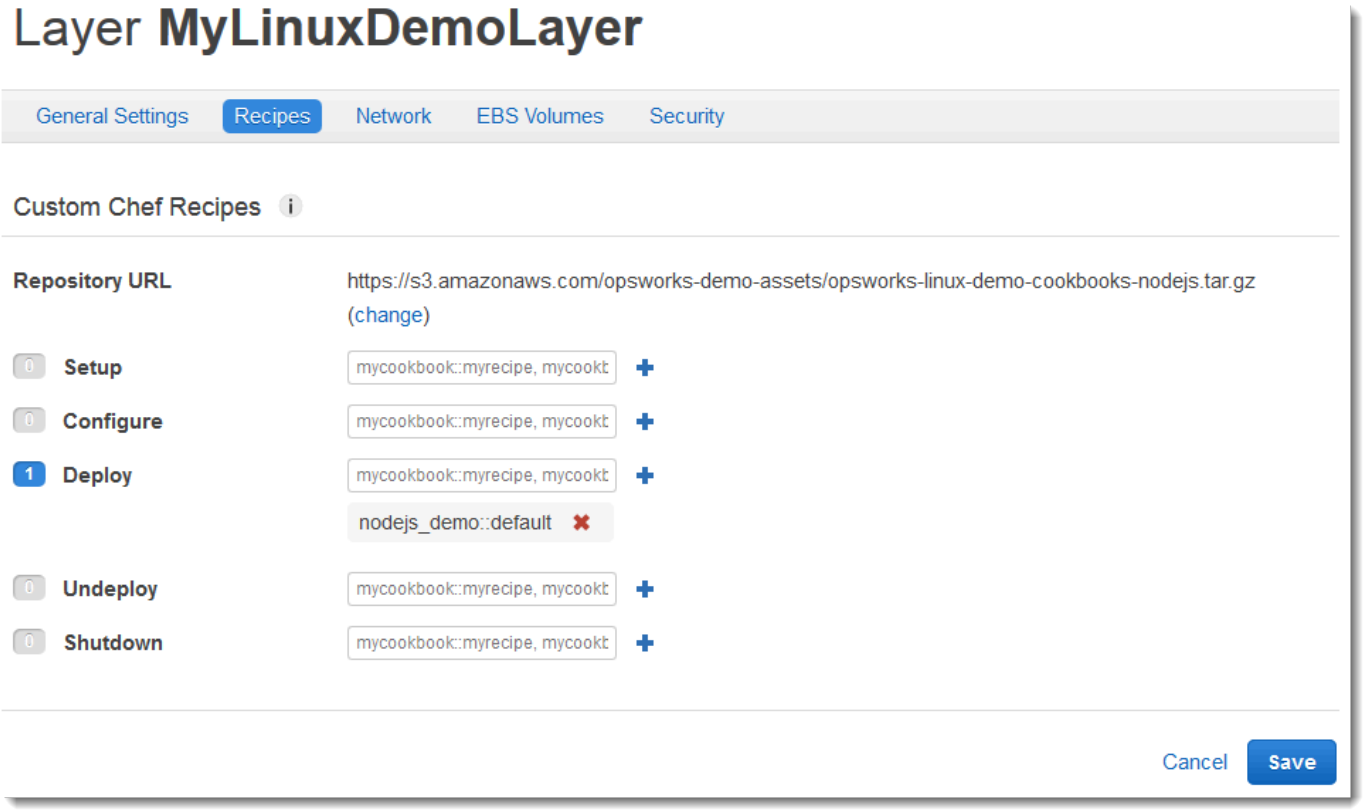
Per specificare la ricetta da utilizzare per distribuire l'applicazione sull'istanza

1. Nel riquadro di navigazione del servizio, scegliere Layers (Livelli). Viene visualizzata la pagina Layers (Livelli).
2. Per MyLinuxDemoLayer, scegli Ricette:



La MyLinuxDemoLayer pagina Layer viene visualizzata con la scheda Ricette aperta.

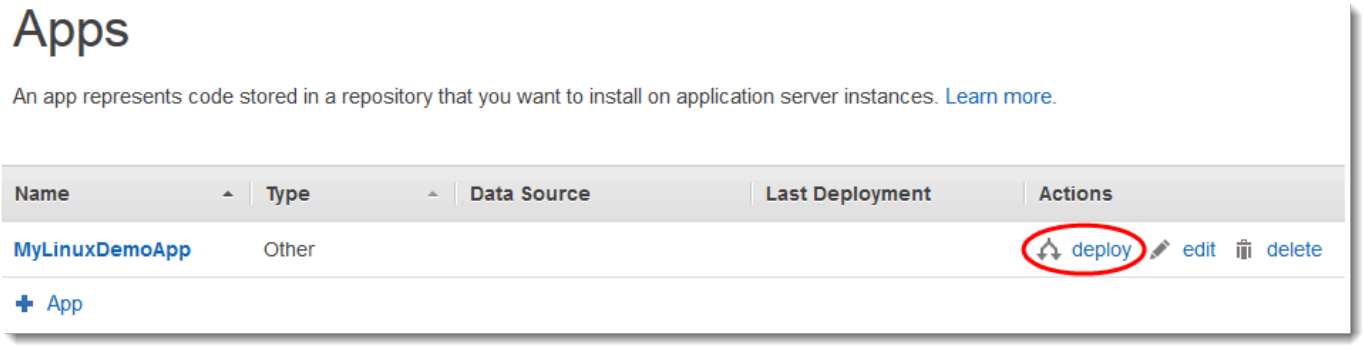
3. Per Custom Chef Recipes (Ricette Chef personalizzate), in Deploy (Distribuzione), digitare **`nodejs_demo::default`** e quindi premere Invio. `nodejs_demo` è il nome del libro di ricette e `default` è il nome della ricetta di destinazione all'interno del libro di ricette. Per esplorare il codice della ricetta, consulta Approfondimento: esplorazione del libro di ricette utilizzato in questa procedura guidata. I risultati devono corrispondere a quelli illustrati nel seguente screenshot:



4. Scegliete Salva. AWS OpsWorks Stacks aggiunge la ricetta all'evento del ciclo di vita Deploy del layer.

Per distribuire l'app nell'istanza

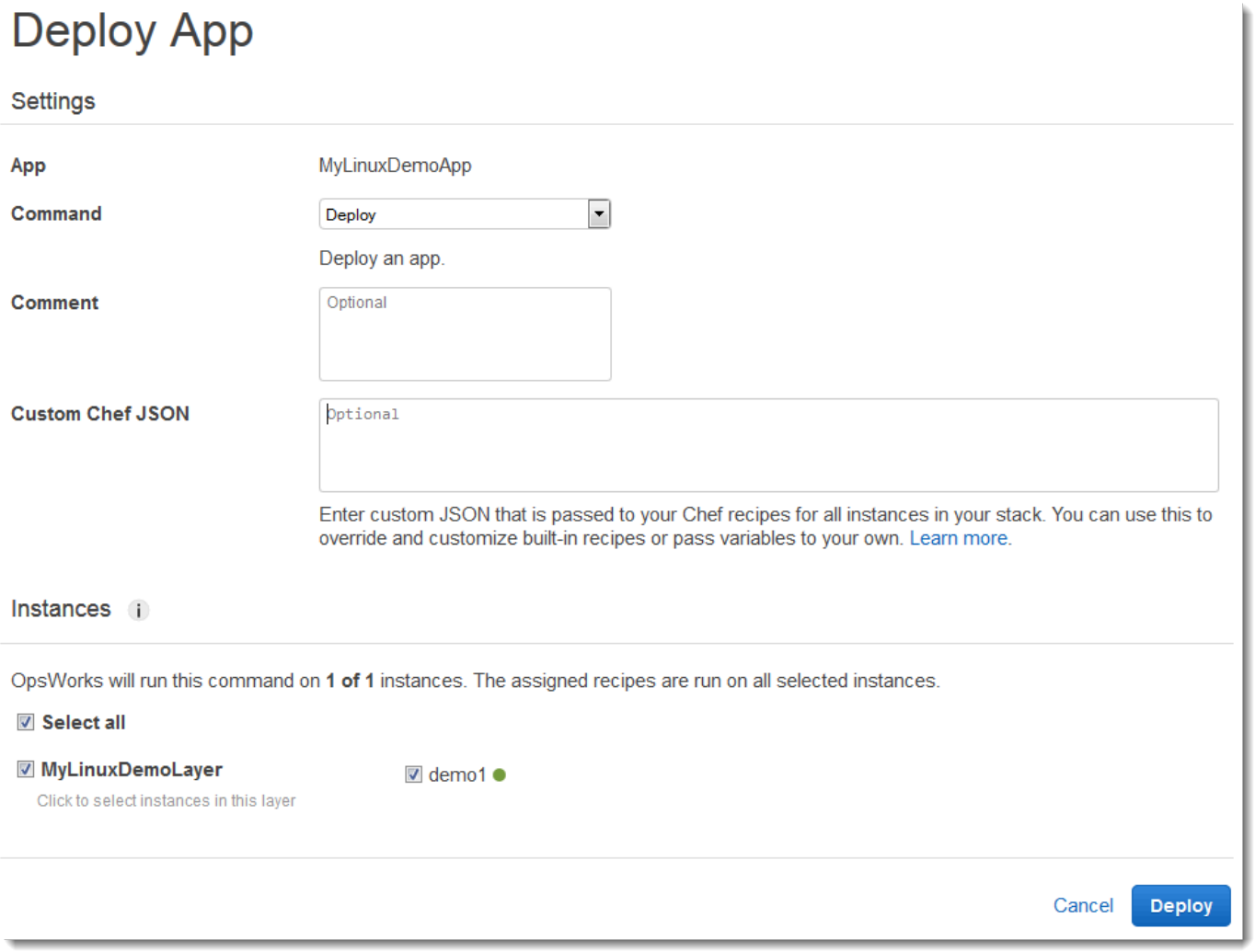
1. Nel riquadro di navigazione del servizio, scegliere Apps (App). Viene visualizzata la pagina Apps (App).
2. Per MyLinuxDemoApp, per Actions, scegliete deploy, come mostrato nella seguente schermata:



3. Nella pagina Deploy App (Distribuisci app) non modificare le impostazioni predefinite delle seguenti opzioni:

   - Command (Comando) (Deploy (Distribuisci))
   - Comment (Commento) (vuoto)
   - Settings (Impostazioni), Advanced (Avanzate), Custom Chef JSON (JSON Chef personalizzato) (vuoto)
   - Istanze, Avanzate (selezionato Seleziona tutto, selezionato MyLinuxDemoLayer, controllato demo1)

4. I risultati devono corrispondere a quelli illustrati nel seguente screenshot:



5. Seleziona Deploy (Implementa). Viene visualizzata la pagina Deployment MyLinuxDemoApp — deploy. Il valore di Status (Stato) cambia da running (esecuzione in corso) a successful (operazione riuscita). Accanto a demo1 viene visualizzato un cerchio in movimento che si trasforma in segno di spunta verde. Questo processo può richiedere alcuni minuti. Procedere solo dopo che in Status (Stato) viene visualizzato il valore successful (operazione riuscita) e solo dopo la visualizzazione dell'icona a forma di segno di spunta verde.
6. I risultati devono corrispondere a quelli illustrati nel seguente screenshot, eccezion fatta per i valori dei campi Created at (Data/ora creazione), Completed at (Data/ora completamento), Duration (Durata) e User (Utente). Se in status (stato) viene visualizzato il valore failed (operazione non riuscita), per eseguire la risoluzione dei problemi, in Log, scegliere show (mostra) per visualizzare i dettagli dell'errore:



A questo punto hai distribuito correttamente l'app sull'istanza.

Nella fase successiva procederai al test dell'app distribuita sull'istanza.

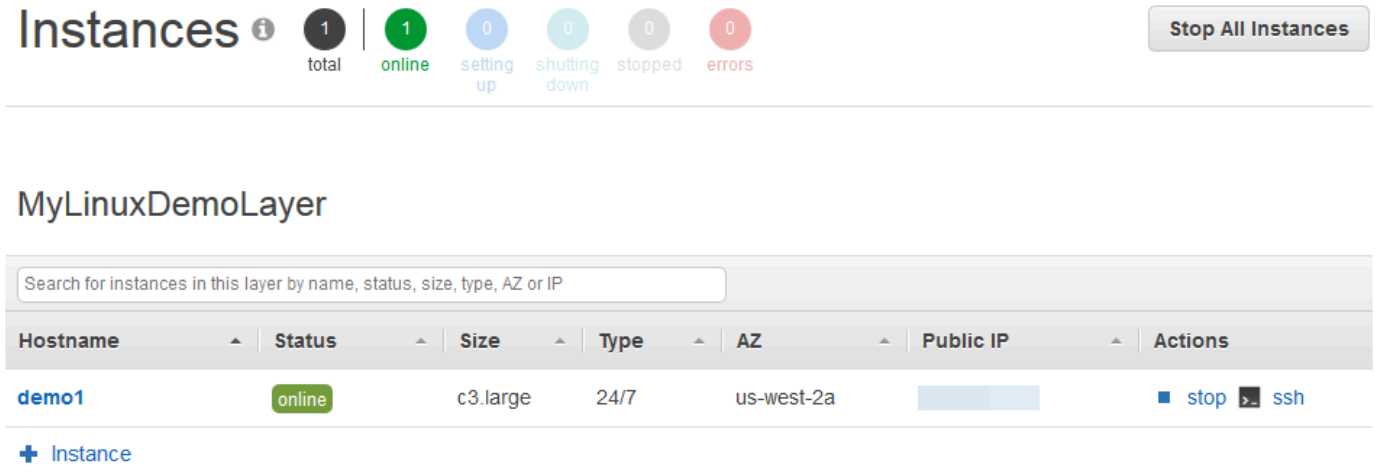
Fase 7. Test dell'app distribuita sull'istanza

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

A questo punto puoi testare la distribuzione sull'istanza.

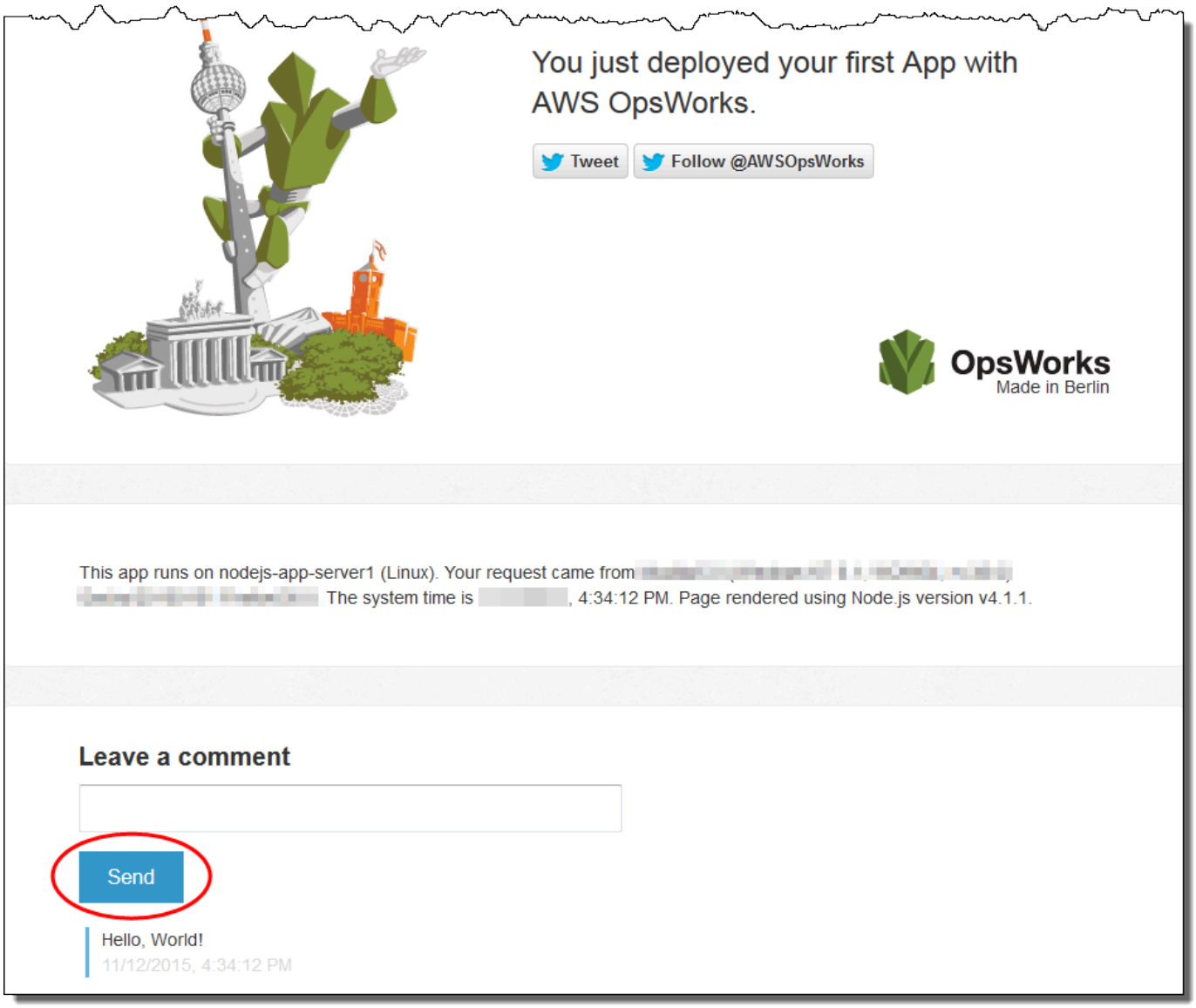
Per testare la distribuzione sull'istanza

1. Nel riquadro di navigazione del servizio, scegliere Instances (Istanze). Viene visualizzata la pagina Instances (Istanze).
2. Per MyLinuxDemoLayer, per demo1, per Public IP, scegli l'indirizzo IP:



L'app viene visualizzata in una nuova scheda del browser Web.

3. Nella pagina Web di presentazione, nella casella di testo Leave a comment (Lascia un commento) immettere un commento, quindi scegliere Send (Invia) per testare l'app. L'app aggiunge il commento alla pagina Web. Continuare a immettere commenti e scegliere Send (Invia):



4. Se hai un account Twitter, scegli Tweet o Segui @ AWSOps Works e segui le istruzioni sullo schermo per twittare sull'app o seguire @ Works. AWSOps

A questo punto hai testato correttamente l'app distribuita sull'istanza.

Nel passaggio successivo, puoi ripulire le AWS risorse utilizzate per questa procedura dettagliata. Questa fase è opzionale. Potresti voler continuare a utilizzare queste AWS risorse man mano che continui a saperne di più su AWS OpsWorks Stacks. Tuttavia, il mantenimento di queste AWS risorse potrebbe comportare alcuni addebiti continui sul tuo AWS account. Se desideri conservare queste AWS risorse per un uso successivo, ora hai completato questa procedura dettagliata e puoi passare alla. Fasi successive

Fase 8. Pulizia (opzionale)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per evitare di incorrere in costi aggiuntivi AWS sul tuo account, puoi eliminare le AWS risorse utilizzate per questa procedura dettagliata. Queste AWS risorse includono lo AWS OpsWorks stack Stacks e i componenti dello stack. (Per ulteriori informazioni, consulta AWS OpsWorks la sezione Prezzi). Tuttavia, potresti voler continuare a utilizzare queste AWS risorse man mano che continui a saperne di più su AWS OpsWorks Stacks. Se desideri mantenere disponibili queste AWS risorse, ora hai completato questa procedura dettagliata e puoi saltare a. Fasi successive

I contenuti archiviati nelle risorse create per questa procedura guidata possono contenere informazioni identificative personali. Se non si desidera che queste informazioni vengano archiviate da AWS, seguire le fasi in questo argomento.

Per eliminare l'app dallo stack

1. Nella console AWS OpsWorks Stacks, nel pannello di navigazione del servizio, scegli App. Viene visualizzata la pagina Apps (App).
2. Per MyLinuxDemoApp, per Azioni, scegli elimina. Quando viene visualizzato il messaggio di conferma, scegli Elimina. AWS OpsWorks Stacks elimina l'app.

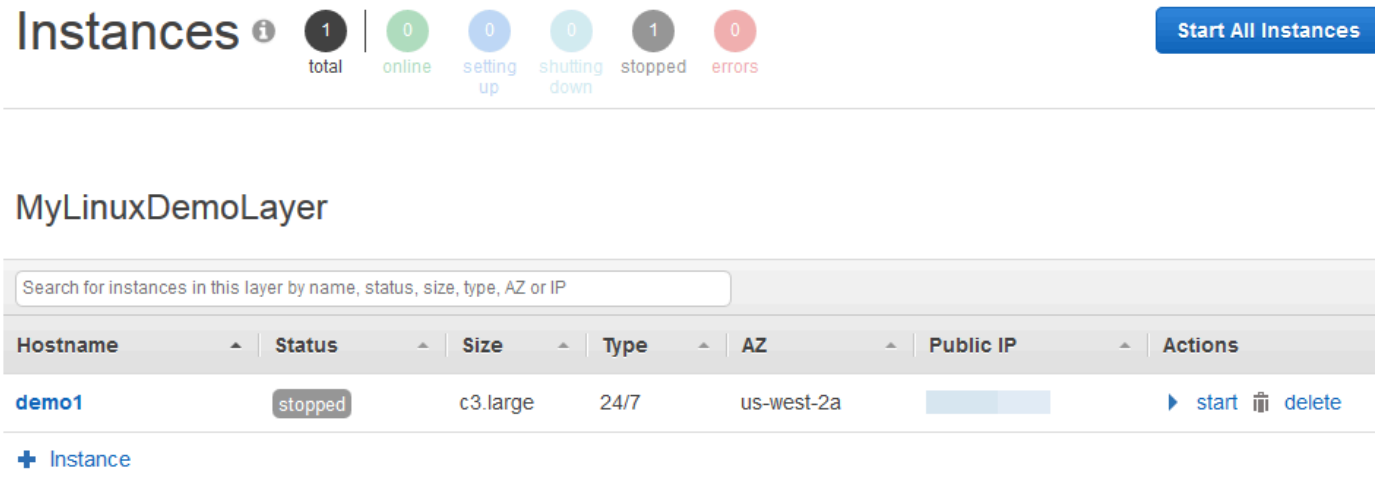
Per eliminare l'istanza per lo stack

1. Nel riquadro di navigazione del servizio, scegliere Instances (Istanze). Viene visualizzata la pagina Instances (Istanze).
2. Per MyLinuxDemoLayer, per demo1, per Actions, scegli stop. Quando viene visualizzato il messaggio di conferma, scegliere Stop (Arresta). Si verificano le seguenti situazioni.

   - L'opzione Status (Stato) passa da online a stopping (arresto in corso) e infine a stopped (arrestato).

- La voce online cambia da 1 in 0.
- La voce shutting down (chiusura in corso) cambia da 0 in 1 e infine in 0.
- La voce stopped (arrestato) cambia da 0 in 1.

Questo processo può richiedere alcuni minuti. Al termine di AWS OpsWorks Stacks, vengono visualizzati i seguenti risultati.



3. In Actions (Operazioni), scegliere delete (eliminazione). Quando viene visualizzato il messaggio di conferma, scegli Elimina. AWS OpsWorks Stacks elimina l'istanza e visualizza il messaggio Nessuna istanza.

Per eliminare lo stack

1. Nel riquadro di navigazione del servizio, scegliere Stack (Stack). Viene visualizzata la pagina MyLinuxDemoStack.
2. Scegliere Delete Stack (Elimina stack). Quando viene visualizzato il messaggio di conferma, scegli Elimina. AWS OpsWorks Stacks elimina lo stack e visualizza la pagina Dashboard. OpsWorks

Facoltativamente, puoi eliminare l'utente e la coppia di EC2 chiavi Amazon che hai usato per questa procedura dettagliata, se non desideri riutilizzarli per accedere ad altri AWS servizi e istanze. EC2 Per istruzioni, consulta Eliminazione di un utente IAM e di coppie di EC2 chiavi Amazon e istanze Linux.

A questo punto hai completato la procedura guidata. Per ulteriori informazioni, consulta Fasi successive.

Fasi successive

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ora che hai completato questa procedura dettagliata, puoi saperne di più sull'uso di Stacks: AWS OpsWorks

- Scopri altre informazioni sul libro di ricette e sull'app utilizzati per questa procedura guidata. Consulta Approfondimento: esplorazione del libro di ricette utilizzato in questa procedura guidata e Approfondimento: esplorazione dell'app utilizzata in questa procedura guidata.
- Esercitati a usare AWS OpsWorks Stacks con istanze Windows. Consultare Nozioni di base: Windows.
- Scopri altre informazioni sugli stack eseguendo l'operazione Creare un nuovo stack.
- Scopri altre informazioni sui livelli eseguendo l'operazione Modifica della configurazione di un livello OpsWorks .
- Scopri altre informazioni sulle istanze eseguendo l'operazione Aggiunta di un'istanza a un livello.
- Scopri altre informazioni sulle app eseguendo l'operazione Distribuzione di app.
- Ulteriori informazioni su Libri di ricette e ricette.
- Crea il tuo libro di ricette. Consultare Nozioni di base: libri di ricette.
- Scopri altre informazioni sul controllo dell'accesso agli stack consultando l'argomento Sicurezza e autorizzazioni.

Approfondimento: esplorazione del libro di ricette utilizzato in questa procedura guidata

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo argomento descrive il ricettario utilizzato da AWS OpsWorks Stacks per la procedura dettagliata.

Un libro di ricette è un concetto Chef. I libri di ricette sono file di archivio contenenti informazioni di configurazione, ad esempio ricette, valori di attributi, file, modelli, librerie, definizioni e risorse personalizzate. Anche la ricetta è un concetto Chef. Le ricette sono istruzioni, scritte con la sintassi del linguaggio Ruby, che specificano le risorse da utilizzare e l'ordine in cui tali risorse vengono applicate. Per ulteriori informazioni, consulta l'argomento relativo all'introduzione dei libri di ricette e all'introduzione delle ricette nel sito Web Learn Chef.

Per visualizzare il contenuto del ricettario utilizzato in questa procedura dettagliata, estrai il contenuto del file opsworks-linux-demo-cookbooks-nodejs.tar.gz in una directory vuota sulla workstation locale. Puoi anche effettuare l'accesso all'istanza in cui hai distribuito il libro di ricette e quindi esplorare i contenuti della directory `/var/chef/cookbooks`.

Il file `default.rb` nella directory `cookbooks/nodejs_demo/recipes` è la posizione in cui il libro di ricette esegue il relativo codice:

```
app = search(:aws_opsworks_app).first
app_path = "/srv/#{app['shortname']}"

package "git" do
  options "--force-yes" if node["platform"] == "ubuntu" && node["platform_version"] ==
 "18.04"
end

application app_path do
  javascript "4"
```

```
  environment.update("PORT" => "80")

  git app_path do
    repository app["app_source"]["url"]
    revision app["app_source"]["revision"]
  end

  link "#{app_path}/server.js" do
    to "#{app_path}/index.js"
  end

  npm_install
  npm_start
end
```

Le operazioni eseguite dal file sono descritte di seguito:

- `search(:aws_opsworks_app).first` utilizza la ricerca di Chef per cercare le informazioni relative all'app che verrà distribuita sull'istanza. Queste informazioni includono impostazioni quali il nome breve dell'app e i dettagli del relativo archivio di origine. Poiché in questa procedura guidata è stata distribuita solo un'app, la ricerca di Chef recupera queste informazioni dal primo elemento dell'indice di ricerca `aws_opsworks_app` sull'istanza. Ogni volta che viene avviata un'istanza, AWS OpsWorks Stacks memorizza questa e altre informazioni correlate come un insieme di data bag sull'istanza stessa e il contenuto del data bag viene visualizzato tramite Chef search. Anche se è possibile eseguire l'hard coding di queste impostazioni in questa ricetta, l'utilizzo dei contenitori di dati e della ricerca di Chef rappresenta l'approccio più affidabile. Per ulteriori informazioni sui contenitori di dati, consulta AWS OpsWorks Riferimento a Stacks Data Bag. Consulta anche l'argomento relativo all'introduzione dei contenitori di dati nel sito Web Learn Chef. Per ulteriori informazioni sulla ricerca di Chef, consulta l'argomento relativo all'introduzione della ricerca nel sito Web Learn Chef.
- La risorsa `package` installa Git sull'istanza.
- La risorsa `application` descrive e distribuisce le applicazioni Web:
  - `javascript`è la versione del JavaScript runtime da installare.
  - `environment` imposta una variabile di ambiente.
  - `git` recupera il codice sorgente dall'archivio e dal ramo specificati.
  - `app_path` è il percorso in cui clonare l'archivio. Se il percorso non esiste nell'istanza, AWS OpsWorks Stacks lo crea.
  - `link` crea un collegamento simbolico.

- `npm_install` installa Node Package Manager, il programma di gestione dei pacchetti predefinito per Node.js.
- `npm_start` esegue Node.js.

Sebbene AWS OpsWorks Stacks abbia creato il ricettario utilizzato per questa procedura dettagliata, puoi creare i tuoi libri di cucina. Per scoprire come, consulta Nozioni di base: libri di ricette. Consulta anche gli argomenti relativi all'introduzione dei libri di ricette, all'introduzione delle ricette e alle nozioni di base su Chef per Ubuntu nel sito Web Learn Chef, nonché la sezione relativa al primo libro di ricette di Chef nell'argomento relativo alle fasi iniziali con Chef nel sito Web Getting started with Chef.

Approfondimento: esplorazione dell'app utilizzata in questa procedura guidata

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo argomento descrive l'app che AWS OpsWorks Stacks distribuisce all'istanza per questa procedura dettagliata.

Per visualizzare il codice sorgente dell'app, estrai il contenuto del opsworks-windows-demo-nodejs GitHub repository in una directory vuota sulla workstation locale. Puoi anche effettuare l'accesso all'istanza in cui hai distribuito il libro di ricette e quindi esplorare i contenuti della directory `/srv/mylinuxdemoapp`.

Il file `index.js` contiene il codice più significativo dell'app:

```
var express = require('express');
var app = express();
var path = require('path');
var os = require('os');
var bodyParser = require('body-parser');
var fs = require('fs');

var add_comment = function(comment) {
```

```
  var comments = get_comments();
  comments.push({"date": new Date(), "text": comment});
  fs.writeFileSync('./comments.json', JSON.stringify(comments));
};

var get_comments = function() {
  var comments;
  if (fs.existsSync('./comments.json')) {
    comments = fs.readFileSync('./comments.json');
    comments = JSON.parse(comments);
  } else {
    comments = [];
  }
  return comments;
};

app.use(function log (req, res, next) {
  console.log([req.method, req.url].join(' '));
  next();
});
app.use(express.static('public'));
app.use(bodyParser.urlencoded({ extended: false }))

app.set('view engine', 'jade');
app.get('/', function(req, res) {
  var comments = get_comments();
  res.render("index",
    { agent: req.headers['user-agent'],
      hostname: os.hostname(),
      nodeversion: process.version,
      time: new Date(),
      admin: (process.env.APP_ADMIN_EMAIL || "admin@unconfigured-value.com" ),
      comments: get_comments()
    });
});

app.post('/', function(req, res) {
  var comment = req.body.comment;
  if (comment) {
    add_comment(comment);
    console.log("Got comment: " + comment);
  }
  res.redirect("/#form-section");
});
```

```
var server = app.listen(process.env.PORT || 3000, function() {
  console.log('Listening on %s', process.env.PORT);
});
```

Le operazioni eseguite dal file sono descritte di seguito:

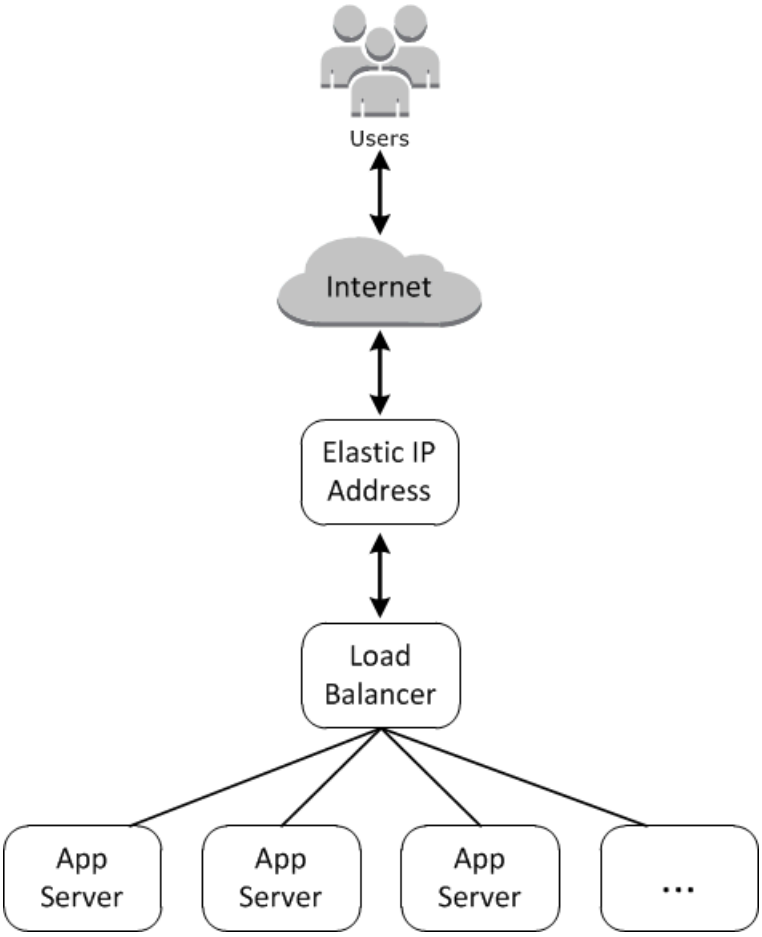
- `require` carica i moduli contenenti parte del codice dipendente necessario per la corretta esecuzione di questa app Web.
- Le funzioni `add_comment` e `get_comments` leggono e scrivono le informazioni nel file `comments.json`.
- Per ulteriori informazioni su `app.get`, `app.listen`, `app.post`, `app.set` e `app.use`, consulta la documentazione di riferimento relativa all'API Express.

Per ulteriori informazioni su come creare l'app e definire il relativo pacchetto per la distribuzione, consulta Origine dell'applicazione.

Nozioni di base sugli Stack Windows

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le applicazioni basate sul cloud di solito richiedono un gruppo di risorse correlate (server di applicazioni, server di database e così via) che devono essere create e gestite collettivamente. Questa raccolta di istanze viene chiamata stack. Un semplice stack di applicazioni potrebbe essere simile al seguente.



L'architettura di base è composta dagli elementi seguenti:

- Un indirizzo IP elastico per ricevere le richieste degli utenti.
- Un sistema di bilanciamento del carico per distribuire le richieste in entrata in modo uniforme in tutti i server di applicazioni.
- Un set di istanze di server di applicazioni, in numero sufficiente a gestire il traffico.

Inoltre, solitamente è necessario un modo per distribuire applicazioni per i server di applicazioni, gestire le autorizzazioni utente e così via.

AWS OpsWorks Stacks offre un modo semplice e diretto per creare e gestire gli stack e le applicazioni e le risorse associate. Questo capitolo introduce le nozioni di base di AWS OpsWorks Stacks, insieme ad alcune delle sue funzionalità più sofisticate, illustrando il processo di creazione dello stack di application server illustrato nel diagramma. Utilizza un modello di sviluppo incrementale che AWS OpsWorks Stacks rende facile da seguire: configura uno stack di base e, dopo che ha funzionato correttamente, aggiungi componenti fino ad arrivare a un'implementazione completa.

- Fase 1: completamento dei prerequisiti mostra come ottenere la configurazione per iniziare la procedura guidata.
- Fase 2: Creare uno stack server di applicazioni di basemostra come creare uno stack di base per supportare Internet Information Services (IIS) e distribuire un'app sul server.
- Fase 3: scalabilità orizzontale IISExample mostra come scalare uno stack per gestire l'aumento di carico aggiungendo più server di applicazioni, un sistema di bilanciamento del carico per distribuire il traffico in entrata e un indirizzo IP elastico per ricevere le richieste in entrata.

Argomenti

- Fase 1: completamento dei prerequisiti
- Fase 2: Creare uno stack server di applicazioni di base
- Fase 3: scalabilità orizzontale IISExample
- Fasi successive

Fase 1: completamento dei prerequisiti

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Prima di poter avviare la procedura guidata, completa le seguenti fasi di configurazione. Questi passaggi di configurazione includono la registrazione di un AWS account, la creazione di un utente amministrativo e l'assegnazione delle autorizzazioni di accesso a Stacks. AWS OpsWorks

Se hai già completato le procedure guidate di Nozioni di base: esempio o di Nozioni di base: Linux, avrai completato i prerequisiti di questa procedura guidata e potrai passare direttamente a Fase 2: Creare uno stack server di applicazioni di base.

Argomenti

- Registrati per un Account AWS

- Crea un utente con accesso amministrativo
- Assegna le autorizzazioni di accesso al servizio
- Assicurati che gli utenti AWS OpsWorks Stacks vengano aggiunti al tuo dominio

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Parte della procedura di registrazione prevede la ricezione di una telefonata o di un messaggio di testo e l'immissione di un codice di verifica sulla tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Assegna le autorizzazioni di accesso al servizio

Abilita l'accesso al servizio AWS OpsWorks Stacks (e ai servizi correlati su cui si basa AWS OpsWorks Stacks) aggiungendo le `AmazonS3FullAccess` autorizzazioni `AWSOpsWorks_FullAccess` e al tuo ruolo o utente.

Per ulteriori informazioni sull'aggiunta di autorizzazioni, consulta Aggiungere i permessi di identità IAM (console).

Assicurati che gli utenti AWS OpsWorks Stacks vengano aggiunti al tuo dominio

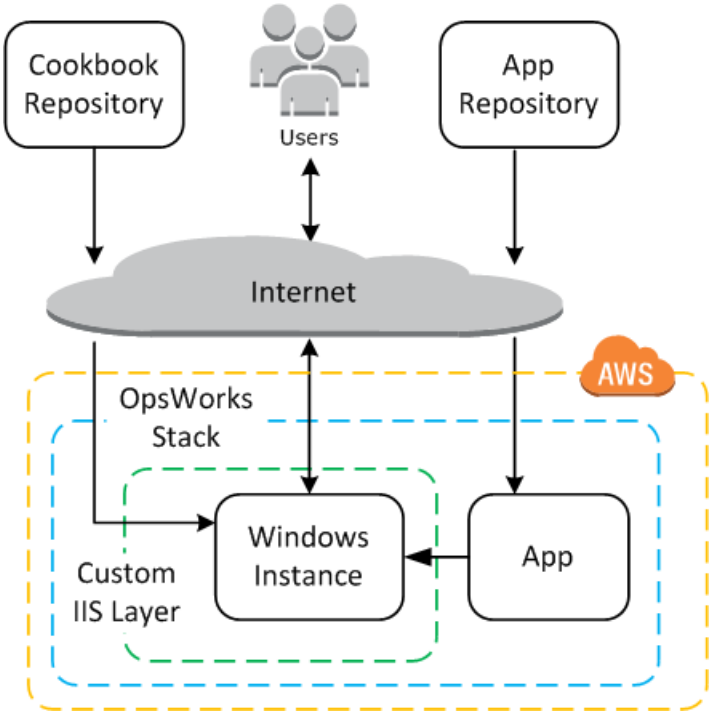
In uno stack di Chef 12.2, il libro di ricette `aws_opsworks_users` incluso crea utenti che dispongono dell'accesso SSH e RDP (Remote Desktop Protocol) alle istanze basate su Windows. Quando unisci istanze Windows del tuo stack a un dominio Active Directory, l'esecuzione di questo ricettario può avere esito negativo se gli utenti AWS OpsWorks Stacks non esistono in Active Directory. Se gli utenti non sono riconosciuti su Active Directory, le istanze possono inserire uno stato `setup failed` quando le riavvii dopo averle unite a un dominio. Per le istanze Windows aggiunte al dominio, non è sufficiente concedere agli utenti di AWS OpsWorks Stacks l'accesso alla pagina delle autorizzazioni utente. SSH/RDP

Prima di aggiungere istanze Windows in uno stack Chef 12.2 a un dominio Active Directory, assicurati che tutti gli utenti AWS OpsWorks Stacks dello stack basato su Windows siano membri del dominio. Il modo migliore per farlo è configurare l'identità federata con IAM prima di creare lo stack basato su Windows, quindi importare gli utenti federati in Stacks prima di unire le istanze dello stack a un dominio. AWS OpsWorks Per ulteriori informazioni su come eseguire questa operazione, consulta Enabling Federation to AWS Using Windows Active Directory, ADFS e SAML 2.0 nel blog di AWS Security e Federating Existing Users nella IAM User Guide.

Fase 2: Creare uno stack server di applicazioni di base

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Uno stack server di applicazioni di base è composto da una singola istanza server di applicazioni e da un indirizzo IP pubblico per ricevere le richieste utente. Il codice dell'applicazione e dei file correlati vengono depositati in un archivio separato e distribuiti da lì al server. Il diagramma seguente illustra tale stack.



Lo stack include i componenti seguenti:

- Un livello, che rappresenta un gruppo di istanze e specifica come devono essere configurate.

  Il livello in questo esempio rappresenta un gruppo di istanze IIS.
- Un'istanza, che rappresenta un' EC2 istanza Amazon.

  In questo caso, il livello configura una singola istanza per eseguire IIS, ma i livelli possono avere un qualsiasi numero di istanze.
- Un' app, che contiene le informazioni necessarie per installare un'applicazione per l'istanza.
- Un libro di ricette, che contiene le ricette personalizzate di Chef che supportano il livello IIS personalizzato. Il ricettario e il codice dell'app vengono archiviati in repository remoti, come un file di archivio in un bucket Amazon S3 o un repository Git.

Le sezioni seguenti descrivono come utilizzare la console AWS OpsWorks Stacks per creare lo stack e distribuire l'applicazione.

Argomenti

- Fase 2.1. Creazione dello stack
- Fase 2.2: autorizzare l'accesso RDP
- Fase 2.3: implementare un libro di ricette personalizzato
- Fase 2.4: aggiungere un livello IIS
- Fase 2.5: Distribuire un'app

Fase 2.1. Creazione dello stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi avviare un progetto AWS OpsWorks Stacks creando uno stack, che funge da contenitore per le tue istanze e altre risorse. La configurazione dello stack specifica alcune impostazioni di base, ad esempio la regione AWS e il sistema operativo predefinito, che sono condivisi da tutte le istanze dello stack.

Per creare un nuovo stack

1. Aggiungi uno stack

   Se non è già stato già fatto, accedere alla console AWS OpsWorks Stacks.

   - Se l'account non ha stack esistenti, vedrai la OpsWorks pagina di benvenuto in AWS; scegli Aggiungi il tuo primo stack.
   - Altrimenti, vedrai la dashboard AWS OpsWorks Stacks, che elenca gli stack del tuo account; scegli Aggiungi stack.
2. Configura lo stack

   Sulla pagina Add Stack (Aggiungi stack), scegli Chef 12 stack (Stack Chef 12) e specifica le impostazioni seguenti:

Stack name (Nome stack)

Inserisci un nome per lo stack, che può contenere caratteri alfanumerici (a—z, A—Z e 0—9) e trattini (-). Il nome dello stack di esempio utilizzato in questa procedura guidata è **`IISWalkthrough`**.

Regione

Seleziona Stati Uniti occidentali (Oregon) come regione dello stack.

Puoi creare uno stack in qualsiasi regione, ma per i tutorial consigliamo US West (Oregon).

Sistema operativo predefinito

Scegli Windows, quindi specifica Microsoft Windows Server 2022 Base, che è l'impostazione predefinita.

Utilizza i libri di ricette personalizzati di Chef

Ai fini di questa procedura guidata, specifica No per questa opzione.

3. Scegli Advanced (Avanzato) per confermare che disponi di un ruolo IAM e del profilo personalizzato dell'istanza IAM selezionata.

   Ruolo IAM

   Specificate il ruolo IAM (AWS Identity and Access Management) dello stack. AWS OpsWorks Stacks deve accedere ad altri servizi AWS per eseguire attività come la creazione e la gestione di EC2 istanze Amazon. Il ruolo IAM specifica il ruolo che AWS OpsWorks Stacks assume per lavorare con altri servizi AWS per tuo conto. Per ulteriori informazioni, consulta Consentire a AWS OpsWorks Stacks di agire per tuo conto.

   - Se il tuo account ha già un ruolo IAM di AWS OpsWorks Stacks, puoi selezionarlo dall'elenco.

     Se il ruolo è stato creato da AWS OpsWorks Stacks, verrà denominato. `aws-opsworks-service-role`
   - Altrimenti, seleziona Nuovo ruolo IAM per indirizzare AWS OpsWorks Stacks a creare un nuovo ruolo per te con le autorizzazioni corrette.

     Nota: se disponi delle autorizzazioni per l'accesso completo su AWS OpsWorks Stacks, la creazione di un nuovo ruolo richiede diverse autorizzazioni IAM aggiuntive. Per ulteriori informazioni, consulta Policy di esempio.

4. Accetta i valori predefiniti per le altre impostazioni, quindi scegli Add Stack (Aggiungi Stack). Per ulteriori informazioni sulle varie impostazioni dello stack, consulta Creare un nuovo stack.

Fase 2.2: autorizzare l'accesso RDP

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver creato uno stack, potrai creare un livello e aggiungere un'istanza di Windows al livello. Tuttavia, prima di farlo, è necessario configurare lo stack per permettere all'utente di utilizzare RDP per connettersi alle istanze di livello personalizzate. Per fare ciò, è necessario completare le attività seguenti:

- Aggiungi una regola in entrata per il gruppo di sicurezza che controlla l'accesso RDP.
- Imposta le autorizzazioni di AWS OpsWorks Stacks per questo stack per consentire l'accesso RDP.

Quando crei il primo stack in una regione, AWS OpsWorks Stacks crea una serie di gruppi di sicurezza. Ne includono uno chiamato qualcosa del genere`AWS-OpsWorks-RDP-Server`, che AWS OpsWorks Stacks collega a tutte le istanze di Windows per consentire l'accesso RDP. Tuttavia, per impostazione predefinita, questo gruppo di sicurezza non presenta regole, perciò devi aggiungere una regola in entrata per consentire l'accesso RDP alle istanze.

Per consentire l'accesso RDP

1. Apri la EC2 console Amazon, impostala sulla regione dello stack e scegli Security Groups dal pannello di navigazione.
2. Scegli AWS- OpsWorks -RDP-Server, scegli la scheda Inbound e scegli Modifica.
3. Scegliere Add Rule (Aggiungi regola) e specificare le seguenti impostazioni:
   - Tipo: RDP.

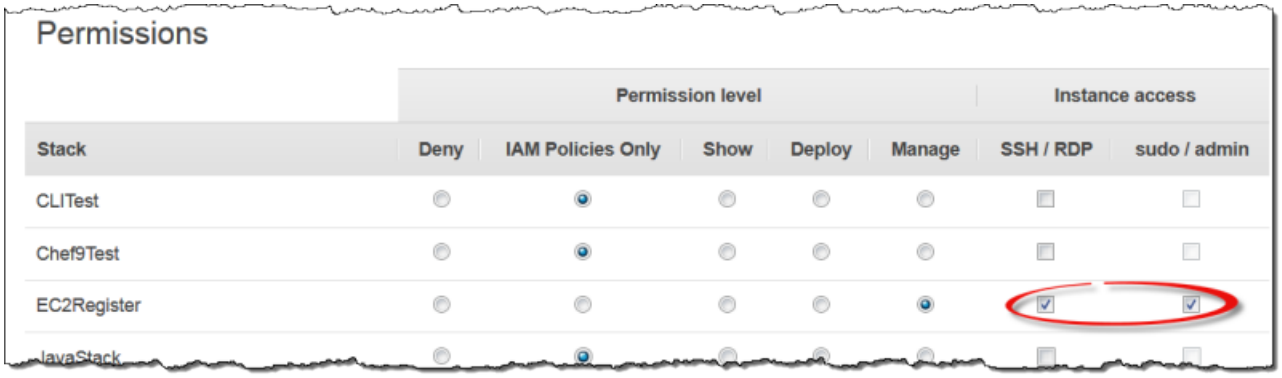
- Fonte: gli indirizzi IP di origine consentiti.

  In genere vengono permesse richieste RDP in entrata dall'indirizzo IP o da un intervallo di indirizzi IP specificato (di solito l'intervallo di indirizzi IP dell'azienda). Ai fini dell'apprendimento, spesso è sufficiente specificare 0.0.0.0/0, che permette l'accesso RDP da qualsiasi indirizzo IP.

Il gruppo di sicurezza permette all'istanza di ricevere richieste di connessione RDP, ma non è tutto. Un utente normale accederà all'istanza utilizzando una password fornita da AWS OpsWorks Stacks. Per fare in modo che AWS OpsWorks Stacks generi quella password, devi autorizzare esplicitamente l'accesso RDP per l'utente.

Per autorizzare l'RDP per un utente

1. Nella dashboard AWS OpsWorks Stacks, scegli lo stack. IISWalkthrough
2. Nel riquadro di navigazione dello stack, scegli Permissions (Autorizzazioni).
3. Nella pagina delle autorizzazioni, scegli Edit (Modifica).
4. Nell'elenco degli utenti, seleziona la casella di controllo SSH/RDP per l'utente a cui desideri concedere le autorizzazioni necessarie. Inoltre, se desideri che l'utente disponga delle autorizzazioni amministratore, seleziona anche sudo/admin .



5. Scegli Save (Salva).

L'utente può quindi ottenere una password e utilizzarla per accedere all'istanza, come descritto più avanti.

> **Note**
>
> Puoi anche accedere come Amministratore. Per ulteriori informazioni, consulta Accesso come amministratore.

Fase 2.3: implementare un libro di ricette personalizzato

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Anche se uno stack è fondamentalmente un contenitore di istanze, non aggiungere istanze direttamente a uno stack. Puoi aggiungere uno o più livelli, ciascuno dei quali rappresenta un gruppo di istanze correlate, quindi aggiungi le istanze ai livelli.

Un layer è fondamentalmente un modello che AWS OpsWorks Stacks utilizza per creare un set di EC2 istanze Amazon con la stessa configurazione. Un'istanza viene avviata con una versione base del sistema operativo, e il livello dell'istanza esegue un'ampia gamma di attività sull'istanza per implementare tale piano, il quale può includere:

- Creazione di directory e file
- Gestione degli utenti
- Installazione e configurazione del software
- Avvio o arresto di server
- Distribuzione del codice applicazione e dei file correlati.

Un layer esegue attività sulle istanze eseguendo ricette Chef, in breve ricette. Una ricetta è un'applicazione di Ruby che utilizza il linguaggio specifico di dominio (DSL) di Chef, per descrivere lo stato finale dell'istanza. Con AWS OpsWorks Stacks, ogni ricetta viene solitamente assegnata a uno degli eventi del ciclo di vita del layer: Setup, Configuration, Deploy, Undeploy e Shutdown. Quando si verifica un evento del ciclo di vita su un'istanza, AWS OpsWorks Stacks esegue le ricette dell'evento

per eseguire le attività appropriate. Ad esempio, l'evento Setup si verifica al termine dell'avvio di un'istanza. AWS OpsWorks Stacks esegue quindi le ricette di installazione, che in genere eseguono attività come l'installazione e la configurazione del software del server e l'avvio dei servizi correlati.

AWS OpsWorks Stacks fornisce a ogni livello una serie di ricette integrate che eseguono attività standard. È quindi possibile estendere una funzionalità del livello per l'implementazione di ricette personalizzate, per eseguire attività aggiuntive e per la loro assegnazione agli eventi del ciclo di vita del livello. Gli stack di Windows supportano livelli personalizzati, i quali dispongono di una serie di ricette che eseguono solo alcune attività di base. Per aggiungere funzionalità alle istanze di Windows, è necessario implementare ricette personalizzate per installare software, distribuire applicazioni e così via. Questo argomento descrive come creare un semplice livello personalizzato per supportare istanze IIS.

Argomenti

- Rapida introduzione ai libri di cucina e alle ricette
- Implementare una ricetta per installare e avviare IIS
- Abilita il libro di ricette personalizzato

Rapida introduzione ai libri di cucina e alle ricette

Una ricetta definisce uno o più aspetti dello stato previsto dell'istanza: di quali directory deve disporre, quali pacchetti software devono essere installati, quali applicazioni devono essere distribuite e così via. Le ricette sono contenute in un libro di ricette, che in genere contiene uno o più ricette correlate, oltre a file associati, ad esempio i modelli per la creazione di file di configurazione.

Questo argomento è un'introduzione molto semplice alle ricette, appena sufficiente per mostrare come implementare un libro di ricette per supportare un semplice livello IIS personalizzato. Per un'introduzione più generale dei libri di ricette, consulta Libri di ricette e ricette. Per un'introduzione tutorial dettagliata relativa all'implementazione di libri di ricette, tra cui alcuni argomenti specifici di Windows, consulta Introduzione ai libri di ricette.

Le ricette di Chef sono tecnicamente applicazioni di Ruby, ma la maggior parte del codice, se non tutto è nel DSL di Chef. Il DSL è composto essenzialmente da una serie di risorse, che è possibile utilizzare per specificare in modo dichiarativo un aspetto dello stato delle istanze. Ad esempio, una `directory` risorsa definisce una directory per essere aggiunta al sistema. L'esempio seguente definisce una `C:\data` directory con i diritti di controllo completi, la quale appartiene all'utente specificato e non eredita i diritti dalla directory padre.

```
directory 'C:\data' do
  rights :full_control, 'WORKGROUP\username'
  inherits false
  action :create
end
```

Quando Chef esegue una ricetta, esegue ogni risorsa passando i dati a un fornitore associato, un oggetto di Ruby che gestisce i dettagli di modifica dello stato dell'istanza. In casi come questo, il fornitore crea una nuova directory con la configurazione specificata.

Il libro di ricette personalizzato per il livello IIS personalizzato deve eseguire le seguenti attività:

- Installare la caratteristica IIS e avviare il servizio.

  Generalmente si esegue questa operazione durante l'installazione, subito dopo il completamento dell'avvio da parte dell'istanza.
- Distribuire un'applicazione per l'istanza, una semplice pagina HTML per questo esempio.

  Generalmente si esegue questa operazione durante l'installazione. Tuttavia, di solito le applicazioni devono essere aggiornati regolarmente, perciò è necessario distribuire gli aggiornamenti quando l'istanza è online.

È possibile disporre di una sola ricetta per eseguire tutte queste attività. Tuttavia, l'approccio consigliato è disporre di ricette separate per l'installazione e le attività di distribuzione. In questo modo, puoi distribuire gli aggiornamenti delle app in qualsiasi momento, senza eseguire anche il codice d'installazione. I seguenti descrivono come configurare un libro di ricette per supportare un livello IIS personalizzato. Gli argomenti di seguito mostreranno come implementare le ricette.

Per iniziare

1. Crea una directory chiamata `iis-cookbook` nella posizione che ti è più comoda sulla workstation.
2. Aggiungi un file `metadata.rb` con il contenuto seguente su `iis-cookbook`.

   ```
   name "iis-cookbook"
   version "0.1.0"
   ```

In questo esempio viene utilizzato un `metadata.rb` minimo. Per ulteriori informazioni su come utilizzare questo file, consulta metadata.rb.

3. Aggiungi una directory `recipes` su `iis-cookbook`.

   Questa directory, che deve essere chiamata `recipes`, contiene le ricette del libro di ricette.

In generale, i libri di ricette possono contenere tante altre directory. Ad esempio, se una ricetta utilizza un modello per creare un file di configurazione, il modello di solito va nella directory `templates\default`. Il libro di ricette di questo esempio, è costituito interamente da ricette, perciò non sono necessarie altre directory. Inoltre, questo esempio utilizza un singolo libro di ricette, ma puoi usarne quanti te ne servono; per progetti complessi sono preferibili più libri di cucina. Ad esempio, è possibile disporre di libri di ricette separati per l'installazione e le attività di distribuzione. Per ulteriori esempi di libri di ricette, consulta Libri di ricette e ricette.

Implementare una ricetta per installare e avviare IIS

IIS è una caratteristica di Windows, uno dei vari componenti di sistema opzionale che è possibile installare su Windows Server. Puoi far sì che una ricetta installi IIS in uno dei seguenti modi:

- Utilizzando una risorsa `powershell_script` per eseguire il cmdlet `Install-WindowsFeature`.
- Utilizzando la risorsa Chef Windows Cookbook`windows_feature`.

  Il libro di ricette di `windows` comprende una serie di risorse, i cui fornitori utilizzano un sistema di gestione e manutenzione delle immagini di distribuzione (DISM) per eseguire varie attività sulle istanze di Windows, tra cui l'installazione della caratteristica.

> **Note**
>
> `powershell_script` è una delle risorse più utili delle ricette di Windows. È possibile utilizzarlo per eseguire diverse attività su un'istanza eseguendo uno PowerShell script o un cmdlet. È utile soprattutto per le attività che non sono supportate da una risorsa di Chef.

Questo esempio esegue uno PowerShell script per installare e avviare Web Server (IIS). Il libro di ricette di `windows` verrà descritto più avanti. Per un esempio su come utilizzare `windows_feature` per installare IIS, consulta Installazione di una funzionalità di Windows: IIS.

Aggiungi una ricetta chiamata `install.rb` sulla directory `recipes` del libro di ricette, con il seguente contenuto.

```
powershell_script 'Install IIS' do
  code 'Install-WindowsFeature Web-Server'
  not_if "(Get-WindowsFeature -Name Web-Server).Installed"
end

service 'w3svc' do
  action [:start, :enable]
end
```

La ricetta contiene due risorse.

powershell_script

`powershell_script`esegue lo PowerShell script o il cmdlet specificato. L'esempio dispone delle seguenti impostazioni dell'attributo:

- `code`— I PowerShell cmdlet da eseguire.

  In questo esempio viene eseguito un cmdlet `Install-WindowsFeature` che installa il server Web (IIS). In generale, l'attributo `code` può avere un numero qualsiasi di linee, così da consentirti di eseguire tutti i cmdlet di cui hai bisogno.
- `not-if`— Un attributo guard che garantisce che la ricetta installi IIS solo se non è ancora stato installato.

  In genere è opportuno che le ricette siano idempotenti, in modo da non perdere tempo per eseguire la stessa attività più di una volta.

Ogni risorsa dispone di un'operazione che specifica l'operazione intrapresa dal fornitore. Non esiste alcuna azione esplicita per questo esempio, quindi il provider esegue l'`:run`azione predefinita, che esegue lo script specificato PowerShell . Per ulteriori informazioni, consulta Esecuzione di uno PowerShell script di Windows.

service

Un `service` gestisce un servizio, il servizio server Web IIS (W3SVC) in questo caso. L'esempio utilizza gli attributi di default e specifica due azioni, `:start` e `:enable`, che avviano e attivano IIS.

> **Note**
>
> Se desideri installare il software che utilizza il programma di installazione del pacchetto, come MSI, puoi utilizzare una risorsa `windows_package`. Per ulteriori informazioni, consulta Installazione di un pacchetto.

Abilita il libro di ricette personalizzato

AWS OpsWorks Stacks esegue le ricette da una cache locale su ogni istanza. Per eseguire le tue ricette personalizzate, è necessario:

- Archiviare il libro di ricette in un repository remoto.

  AWS OpsWorks Stacks scarica i libri di cucina da questo repository nella cache locale di ogni istanza.
- Modificare lo stack per abilitare i libri di ricette personalizzati.

  I libri di ricette personalizzati sono disabilitati per impostazione predefinita; per questo è necessario abilitarli per lo stack e fornire l'URL dell'archivio e delle informazioni correlate.

AWS OpsWorks Stacks supporta archivi S3 e repository Git per libri di cucina personalizzati; questo esempio utilizza un archivio S3. Per ulteriori informazioni, consulta Repository dei libri di ricette.

Per utilizzare un archivio S3

1. Creare un archivio `.zip` della directory `iis-cookbook`.

   AWS OpsWorks Stacks supporta anche archivi `.tgz` (gzip compressi tar) per gli stack di Windows.
2. Carica l'archivio in un bucket S3 nella regione Stati Uniti occidentali (California settentrionale) e rendi il file pubblico. Puoi anche utilizzare un archivio S3 privato, ma un archivio pubblico è sufficiente in questo esempio, ed è anche molto più facile da utilizzare.

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

   b. Se non hai ancora un bucket`us-west-1`, scegli Crea bucket e crea un bucket nella regione Stati Uniti occidentali (California settentrionale).

c. Nell'elenco dei bucket, scegli il nome del bucket sul quale desideri caricare il file, quindi fai clic su Upload (Carica).

d. Seleziona Aggiungi file.

e. Selezionare l'archivio di file da caricare, quindi scegliere Apri.

f. Nella parte inferiore della finestra di dialogo Upload - Select Files and Folders (Carica - Seleziona file e cartelle), scegli Set Details (Imposta dettagli).

g. Nella parte inferiore della finestra di dialogo Set Details (Imposta dettagli), scegli Set Permissions (Imposta autorizzazioni).

h. Nella finestra di dialogo Set Permissions (Imposta autorizzazioni), scegli Make everything public (Rendi tutto pubblico).

i. Nella parte inferiore della finestra di dialogo Set Permissions (Imposta autorizzazioni), scegli Start Upload (Avvia caricamento). Al termine del caricamento, il file `iis-cookbook.zip` viene visualizzato nel bucket.

j. Scegli il bucket e quindi scegli la scheda Proprietà per il bucket. Accanto al Link (Collegamento), registra l'archivio URL del file da utilizzare in seguito.

Per ulteriori informazioni sul caricamento di file in un bucket Amazon S3, consulta Come si caricano file e cartelle in un bucket S3? nella Guida per l'utente della console Amazon S3.

> **Important**
>
> Fino a questo punto, la procedura guidata ti è costata solo un po' di tempo; lo stesso servizio AWS OpsWorks Stacks è gratuito. Tuttavia, devi pagare per tutte le risorse AWS che utilizzi, come lo storage Amazon S3. Ti verranno addebitati costi solo al caricamento dell'archivio. Per ulteriori informazioni, consulta Prezzi di AWS.

Per abilitare libri di ricette personalizzati per lo stack

1. Nella console AWS OpsWorks Stacks, scegli Stack nel pannello di navigazione, quindi scegli Stack Settings in alto a destra.
2. Sulla pagina Settings (Impostazioni), scegli Edit (Modifica) in alto a destra.
3. Sulla pagina Settings (Impostazioni), imposta Use custom Chef cookbooks (Utilizza libri di ricette personalizzati di Chef) su Yes (Sì), quindi inserisci le informazioni seguenti:

- Tipo di repository: S3 Archive.
- URL del repository: l'URL S3 del file di archivio del libro di cucina che hai registrato in precedenza.

4. Scegli Save (Salva) per aggiornare la configurazione dello stack.

AWS OpsWorks Stacks installa il tuo ricettario personalizzato su tutte le nuove istanze. Nota che AWS OpsWorks Stacks non installa automaticamente né aggiorna i libri di ricette personalizzati sulle istanze online. Come descritto in seguito, puoi farlo manualmente.

Fase 2.4: aggiungere un livello IIS

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Il tuo libro di ricette dispone di una ricetta che non fa altro che installare e avviare IIS. Questo è sufficiente per creare il livello e verificare che disponi di un'istanza IIS funzionante. In seguito, potrai aggiungere la funzionalità di distribuzione dell'applicazione al livello.

Creare un livello

Puoi iniziare aggiungendo un layer allo stack. Puoi dunque aggiungere funzionalità per quel livello assegnando ricette personalizzate agli eventi appropriati del ciclo di vita.

Per aggiungere un livello IIS allo stack

1. Scegli Layers (Livelli) nel riquadro di navigazione, quindi scegli Add a layer (Aggiungi un livello).
2. Configura il livello come segue:
   - Nome — **`IISExample`**
   - Nome breve — **`iisexample`**

AWS OpsWorks Stacks utilizza il nome breve per identificare internamente il livello. Puoi anche utilizzare il nome breve per identificare il livello nelle ricette, anche se questo esempio non lo fa. È possibile specificare qualsiasi nome breve, ma possono contenere solo caratteri alfanumerici minuscoli e un piccolo numero di segni di interpunzione. Per ulteriori informazioni, consulta Livelli personalizzati.

3. Scegli Add Layer (Aggiungi livello).

Se a questo IISWalkthrough punto aggiungessi un'istanza e la avviassi, AWS OpsWorks Stacks installerebbe automaticamente i libri di cucina ma non funzionerebbe. `install.rb` Dopo che un'istanza è online, puoi eseguire manualmente le ricette manualmente utilizzando il comando di esecuzione dello stack delle ricette. Tuttavia, un approccio migliore consiste nell'assegnare la ricetta a uno degli eventi del ciclo di vita del layer. AWS OpsWorks Stacks esegue quindi automaticamente la ricetta nel punto appropriato del ciclo di vita dell'istanza.

Installa e avvia IIS non appena viene completato il processo di avvio dell'istanza. Per eseguire questa operazione, assegna `install.rb` all'evento `Setup` del livello.

Per assegnare la ricetta all'evento del ciclo di vita

1. Nel riquadro di navigazione, scegli Layers (Livelli)
2. Nella casella relativa al IISExamplelivello, scegli Ricette.
3. In alto a destra, scegli Edit (Modifica).
4. In Custom Chef Recipes (Ricette di Chef personalizzate), nella casella di ricette Setup (Installazione), digitare **`iis-cookbook::install`**.

> **Note**
>
> Utilizza *`cookbook-name`*`::`*`recipe-name`* per identificare le ricette in cui viene omesso il suffisso `.rb` del nome della ricetta.

5. Scegli + per aggiungere la ricetta al livello. Una x rossa apparirà vicino alla ricetta, affinché sia ancora più semplice rimuoverla in seguito.
6. Scegli Save (Salva) per salvare la nuova configurazione. Le ricette d'installazione personalizzate includeranno ora `iis-cookbook::install`.

Aggiungi un'istanza a un livello e avviala

Puoi provare la ricetta aggiungendo un'istanza al livello e avviando l'istanza. AWS OpsWorks Stacks installa automaticamente i libri di cucina e viene eseguito `install.rb` durante la configurazione, non appena l'avvio dell'istanza termina.

Per aggiungere un'istanza al livello e avviarla

1. Nel pannello di navigazione AWS OpsWorks Stacks, scegli Istanze.
2. In IISExampleLivello, scegli Aggiungi un'istanza.
3. Selezionare la dimensione appropriata. t2.micro (o la dimensione minima disponibile) deve essere sufficiente per l'esempio.
4. Scegliere Add Instance (Aggiungi istanza). Per impostazione predefinita, AWS OpsWorks Stacks genera i nomi delle istanze aggiungendo un numero intero al nome breve del livello, quindi l'istanza deve essere denominata iisexample1.
5. Scegli start nella colonna Azioni dell'istanza per avviare l'istanza. AWS OpsWorks Stacks avvierà quindi un' EC2 istanza ed eseguirà le ricette di installazione per configurarla. Se il layer avesse delle ricette Deploy a questo punto, AWS OpsWorks Stacks le eseguirà al termine delle ricette di installazione.

   Il processo potrebbe richiedere alcuni minuti, durante i quali la colonna Status (Stato) mostra una serie di stati di stato. Quando ottieni lo stato online, il processo d'installazione è completo e l'istanza è pronta all'uso.

Verifica che IIS sia installato e che stia eseguendo

Puoi utilizzare RDP per connetterti all'istanza e verificare che la ricetta d'installazione stia funzionando correttamente.

Per verificare che IIS sia installato e che stia eseguendo

1. Scegli Istanze nel riquadro di navigazione e scegli rdp nella colonna Azioni dell'istanza iisexample1. AWS OpsWorks Stacks genera automaticamente una password RDP che scade dopo un periodo di tempo specificato.
2. Imposta l'elemento Session valid for (Sessione valida per) a due ore e scegli Generate Password (Genera password).

3. AWS OpsWorks Stacks visualizza la password e anche, per comodità, il nome DNS pubblico e il nome utente dell'istanza. Copia tutte e tre le voci, quindi fai clic su Acknowledge and close (Riconosci e chiudi).
4. Apri il client RDP e utilizza i dati della Fase 3 per connetterti all'istanza.
5. Sull'istanza, apri Windows Explorer e esamina l'unità `C:`. Deve disporre di una directory `C:\inetpub`, che è stata creata per l'installazione di IIS.
6. Apri l'applicazione Administrative Tools (Strumenti di amministrazione) del Pannello di controllo, quindi apri Services (Servizi). Dovresti visualizzare il servizio IIS nella parte inferiore dell'elenco. È chiamato World Wide Web Publishing Service, e lo stato deve essere running (in esecuzione).
7. Torna alla console AWS OpsWorks Stacks e scegli l'indirizzo IP pubblico dell'istanza iisexample1. Assicurati di farlo in AWS OpsWorks Stacks e non nella EC2 console Amazon. Questa operazione invia automaticamente una richiesta HTTP all'indirizzo, che apre la pagina di benvenuto di IIS predefinita.

Il prossimo argomento illustra come distribuire un'applicazione per l'istanza, una semplice pagina HTML statica per questo esempio. Tuttavia, se vuoi prendere una pausa, scegli stop (interrompi) nella colonna Actions (Azioni) dell'istanza iisexample1 per interrompere l'istanza ed evitare che vengano addebitati costi non necessari. Puoi riavviare l'istanza quando sei pronto per continuare.

Fase 2.5: Distribuire un'app

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

L'installazione di IIS crea una directory `C:\inetpub\wwwroot` per il codice dell'applicazione e i relativi file. La prossima fase riguarda l'installazione di un'app nella suddetta directory. Per questo esempio installerai una home page HTML statica `default.html`, su `C:\inetpub\wwwroot`. Puoi estendere facilmente l'approccio generale per gestire scenari più complessi, ad esempio le applicazioni ASP.NET.

Puoi includere i file dell'applicazione nei tuoi libri di ricette e copiare `install.rb` su `C:\inetpub\wwwroot`. Per esempi su come eseguire questa operazione, consulta Esempio 6: creazione di file. Tuttavia, questo approccio non è molto flessibile né efficiente, ed è in genere preferibile lo sviluppo di un libro di ricette separato rispetto allo sviluppo dell'applicazione.

La soluzione preferita consiste nell'implementare una ricetta di distribuzione separata che recuperi il codice dell'applicazione e i file correlati da un repository, qualsiasi repository tu preferisca, non solo il repository dei libri di cucina, e lo installa su ogni istanza del server IIS. Questo approccio separa lo sviluppo di libri di ricette da quello dell'applicazione, e quando devi aggiornare l'app, ti permette di eseguire nuovamente solo la ricetta di distribuzione, senza dover aggiornare i libri di ricette.

Questo argomento mostra come implementare una semplice ricetta di distribuzione, la quale distribuisce `default.htm` per il server IIS. È possibile estendere questo esempio per le applicazioni più complesse.

Argomenti

- Creare l'applicazione e archiviarla
- Implementare una ricetta per distribuire l'applicazione
- Aggiornare il libro di ricette dell'istanza
- Aggiungere la ricetta al livello IIS personalizzato
- Aggiungere un'app
- Distribuire l'app e aprire l'applicazione

Creare l'applicazione e archiviarla

Puoi utilizzare qualsiasi archivio per le tue applicazioni. Per semplicità, questo esempio archivia `default.htm` in un bucket S3 pubblico.

Per creare l'applicazione

1. Crea una directory chiamata `iis-application` nella posizione che ti è più comoda sulla workstation.
2. Aggiungere un file `default.htm` a `iis-application` con il contenuto seguente.

```
<!DOCTYPE html>
<html>
  <head>
    <title>IIS Example</title>
```

```
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

3. Creare un bucket S3, caricare `default.htm` nel bucket e registrare l'URL per utilizzarlo successivamente. Per semplicità, rendere pubblico il file.

> **Note**
>
> Si tratta di una semplice applicazione, ma puoi estendere i principi di base per gestire le applicazioni a livello di produzione.
>
> - Per ulteriori applicazioni complesse con più file, di solito è più semplice creare un archivio .zip di `iis-application` e caricarlo sul tuo bucket S3.
>
>   Puoi ora scaricare il file .zip e estrarre i contenuti nella directory appropriata. Non è necessario scaricare più file, né creare una struttura di directory e così via.
> - Per un'applicazione di produzione, vorrai probabilmente mantenere privati i tuoi file. Per vedere un esempio di come una ricetta scarica file da un bucket S3 privato, consulta Utilizzo dell'SDK for Ruby AWS OpsWorks su un'istanza Stacks Windows.
> - Puoi archiviare la tua applicazione in tutti gli archivi idonei.
>
>   L'applicazione si scarica generalmente utilizzando un'API pubblica dell'archivio. Questo esempio utilizza l'API Amazon S3. Se, ad esempio, memorizzi l'applicazione su GitHub, puoi utilizzare l'GitHub API.

Implementare una ricetta per distribuire l'applicazione

Aggiungi una ricetta chiamata `deploy.rb` alla directory `iis-cookbook` `recipes` con i seguenti contenuti.

```
chef_gem "aws-sdk-s3" do
  compile_time false
  action :install
end

ruby_block "download-object" do
```

```
  block do
    require 'aws-sdk-s3'

    #1
    # Aws.config[:ssl_ca_bundle] = 'C:\ProgramData\Git\bin\curl-ca-bundle.crt'
    Aws.use_bundled_cert!

    #2
    query = Chef::Search::Query.new
    app = query.search(:aws_opsworks_app, "type:other").first
    s3region = app[0][:environment][:S3REGION]
    s3bucket = app[0][:environment][:BUCKET]
    s3filename = app[0][:environment][:FILENAME]

    #3
    s3_client = Aws::S3::Client.new(region: s3region)
    s3_client.get_object(bucket: s3bucket,
                         key: s3filename,
                         response_target: 'C:\inetpub\wwwroot\default.htm')
  end
  action :run
end
```

Questo esempio utilizza SDK for Ruby v2 per scaricare il file. Tuttavia, AWS OpsWorks Stacks non installa questo SDK su istanze Windows, quindi la ricetta inizia con `chef_gem`resource, che gestisce tale attività.

> **Note**
>
> La risorsa `chef_gem` installa gem alla versione Ruby dedicata di Chef, che è la versione utilizzata dalle ricette. Se desideri installare una gem per una versione Ruby a livello di sistema, utilizza la risorsa gem_package.

La maggior parte della ricetta è una `ruby_block`risorsa, che esegue un blocco di codice Ruby che utilizza l'SDK for Ruby per il download. `default.htm` Il codice di `ruby_block` può essere suddiviso nelle seguenti sezioni, che corrispondono ai commenti numerati nel codice esempio.

1: Specificare un bundle di certificati.

Amazon S3 utilizza SSL, quindi è necessario un certificato appropriato per scaricare oggetti da un bucket S3. SDK for Ruby v2 non include un pacchetto di certificati, quindi è necessario fornirne

uno e configurare l'SDK per Ruby per utilizzarlo. AWS OpsWorks Stacks non installa direttamente un pacchetto di certificati, ma installa Git, che include un pacchetto di certificati (). `curl-ca-bundle.crt` Per comodità, questo esempio configura l'SDK for Ruby per utilizzare il pacchetto di certificati Git per SSL. Inoltre, puoi installare il tuo bundle e configurare l'SDK di conseguenza.

2: Recuperare i dati dell'archivio

Per scaricare un oggetto da Amazon S3, sono necessari la regione AWS, il nome del bucket e il nome della chiave. Come descritto più avanti, l'esempio fornisce queste informazioni associando una serie di variabili ambiente con l'app. Quando distribuisci un'app, AWS OpsWorks Stacks aggiunge un set di attributi all'oggetto nodo dell'istanza. Questi attributi sono essenzialmente una tabella hash che contiene la configurazione dell'app, tra cui le variabili ambiente. Gli attributi dell'app per questa applicazione saranno simili ai seguenti, in formato JSON.

```
{
  "app_id": "8f71a9b5-de7f-451c-8505-3f35086e5bb3",
  "app_source": {
      "password": null,
      "revision": null,
      "ssh_key": null,
      "type": "other",
      "url": null,
      "user": null
  },
  "attributes": {
      "auto_bundle_on_deploy": true,
      "aws_flow_ruby_settings": {},
      "document_root": null,
      "rails_env": null
  },
  "data_sources": [{"type": "None"}],
  "domains": ["iis_example_app"],
  "enable_ssl": false,
  "environment": {
      "S3REGION": "us-west-2",
      "BUCKET": "windows-example-app",
      "FILENAME": "default.htm"
  },
  "name": "IIS-Example-App",
  "shortname": "iis_example_app",
  "ssl_configuration": {
      "certificate": null,
      "private_key": null,
```

```
      "chain": null
  },
  "type": "other",
  "deploy": true
}
```

Le variabili ambiente dell'app vengono archiviate nell'attributo `[:environment]`. Per recuperarli, utilizza una query di ricerca Chef per recuperare la tabella hash dell'app che si trova nel nodo `aws_opsworks_app`. Quest'app verrà definita come tipo `other`, così che la query cerchi le app di questo tipo. La ricetta sfrutta il fatto che esiste una sola app su questa istanza, perciò la tabella hash di interesse è solo `app[0]`. Per comodità, la ricetta assegna quindi la regione, il bucket e i nomi dei file alle variabili.

Per ulteriori informazioni sull'utilizzo della ricerca Chef, consulta Ottenimento dei valori degli attributi con la ricerca di Chef.

3: Scaricare il file.

La terza parte della ricetta crea un oggetto del client S3 e utilizza il relativo metodo `get_object` per scaricare `default.htm` nella directory `C:\inetpub\wwwroot` dell'istanza.

> **Note**
>
> Un ricetta è un'applicazione Ruby, quindi il codice di Ruby non deve trovarsi necessariamente in un `ruby_block`. Tuttavia, il codice nel corpo della ricetta viene eseguito per primo, seguito in ordine dalle risorse. Per questo esempio, se inserisci il codice di download nel corpo della ricetta, non riuscirebbe perché la `chef_gem` risorsa non avrebbe ancora installato l'SDK for Ruby. Il codice nella `ruby_block` risorsa viene eseguito quando la risorsa viene eseguita, dopo che la `chef_gem` risorsa ha installato l'SDK for Ruby.

Aggiornare il libro di ricette dell'istanza

AWS OpsWorks Stacks installa automaticamente libri di cucina personalizzati su nuove istanze. Tuttavia, se stai utilizzando un'istanza esistente, devi aggiornare il tuo libro di ricette manualmente.

Per aggiornare i libri di ricette dell'istanza

1. Crea un archivio `.zip` di `iis-cookbook` e caricalo sul bucket S3.

Questa operazione sovrascrive il libro di ricette esistente, ma l'URL rimane invariato, perciò non è necessario aggiornare la configurazione dello stack.

2. Riavvia l'istanza se non è online.
3. Dopo che l'istanza è online, scegli Stack nel riquadro di navigazione, quindi scegli Run Command (Esegui comando).
4. In Command (Comando), scegliere Update Custom Cookbooks (Aggiorna libri di ricette personalizzati). Questo comando installa il libro di ricette aggiornato sull'istanza.
5. Scegli Update Custom Cookbooks (Aggiorna libri di ricette personalizzati). L'esecuzione del comando potrebbe richiedere alcuni minuti.

Aggiungere la ricetta al livello IIS personalizzato

Come per `install.rb`, il modo migliore per gestire la distribuzione è quello di assegnare `deploy.rb` all'evento ciclo di vita appropriato. Generalmente le ricette di distribuzione vengono assegnate all'evento Distribuzione e vengono definite collettivamente ricette di Distribuzione. L'assegnazione di una ricetta all'evento Distribuzione non attiva l'evento. Invece:

- Per le nuove istanze, AWS OpsWorks Stacks esegue automaticamente le ricette Deploy al termine delle ricette di installazione, in modo che le nuove istanze abbiano automaticamente la versione corrente dell'applicazione.
- Per le istanze online, utilizza un comando di distribuzione per installare manualmente applicazioni nuove o aggiornate.

  Questo comando attiva un evento Distribuzione sulle istanze dello stack, che esegue le ricette Distribuzione.

Per assegnare deploy.rb all'evento Distribuzione del livello

1. Scegli Livelli nel pannello di navigazione, quindi scegli Ricette in Livello. IISExample
2. In Custom Chef Recipes (Ricette di Chef personalizzate), aggiungere **`iis-cookbook::deploy`** alla casella di ricette Deploy (Distribuisci) e scegliere + per aggiungere la ricetta al livello.
3. Scegli Save (Salva) per salvare la nuova configurazione. Le ricette Distribuzione personalizzate includeranno ora `iis-cookbook::deploy`.

Aggiungere un'app

L'ultima operazione consiste nell'aggiungere un'app allo stack per rappresentare l'applicazione nell'ambiente AWS OpsWorks Stacks. Un'app include i metadati, ad esempio il nome di visualizzazione dell'applicazione e i dati necessari per scaricare l'app dal suo archivio.

Per aggiungere l'app allo stack

1. Scegli Apps (App) nel riquadro di navigazione, quindi scegli Add an app (Aggiungi un'app).
2. Configura l'app con le seguenti impostazioni.
   - Nome: l **`IIS-Example-App`**
   - Tipo di repository: Altro
   - Variabili di ambiente: aggiungere le seguenti tre variabili di ambiente:
     - **`S3REGION`**— La regione del bucket (in questo caso,`us-west-1`).
     - **`BUCKET`**— Il nome del bucket, ad esempio. `windows-example-app`
     - **`FILENAME`**— Il nome del file:**`default.htm`**.
3. Accetta i valori predefiniti delle impostazioni restanti, quindi scegli Add App (Aggiungi app) per aggiungere l'app allo stack.

> **Note**
>
> Questo esempio utilizza variabili ambiente per fornire i dati del download. Un approccio alternativo consiste nell'utilizzare un tipo di repository S3 Archive e fornire l'URL del file. AWS OpsWorks Stacks aggiunge le informazioni, insieme a dati opzionali, come le credenziali AWS, all'attributo dell'`app_source`app. Per estrarre la regione, il nome del bucket e il nome di file. la ricetta distribuzione deve ottenere l'URL dagli attributi dell'app.

Distribuire l'app e aprire l'applicazione

AWS OpsWorks Stacks distribuisce automaticamente le app su nuove istanze, ma non su istanze online. Poiché l'istanza è già in esecuzione, è necessario distribuire l'app manualmente.

Per distribuire l'app

1. Scegli Apps (App) nel riquadro di navigazione, quindi scegli deploy (distribuzione) nella colonna dell'app Actions (Azioni).
2. Il campo Command (Comando) deve essere impostato su Deploy (Distribuisci). Scegli Deploy (Distribuisci) nella parte inferiore destra della pagina Deploy App (Distribuisci app). L'esecuzione del comando potrebbe richiedere alcuni minuti.

   Al termine della distribuzione, viene nuovamente visualizzata la pagina App. L'indicatore Status (Stato) mostra successful (completato) in verde, quindi accanto al nome dell'app appare una spunta verde di verifica per indicare che la distribuzione è stata completata con successo.

> **Note**
>
> Le app di Windows sono sempre di tipo Other (Altro), quindi distribuire l'app apporta le operazioni seguenti:
>
> - Aggiunge i dati dell'app alla configurazione dello stack e agli attributi di distribuzione, come descritto in precedenza.
> - Attiva l'evento Distribuzione sulle istanze dello stack, che esegue le ricette Distribuzione personalizzate.

> **Note**
>
> Per ulteriori informazioni su come risolvere i problemi relativi alle mancate distribuzioni o applicazioni, consulta Il debug delle ricette.

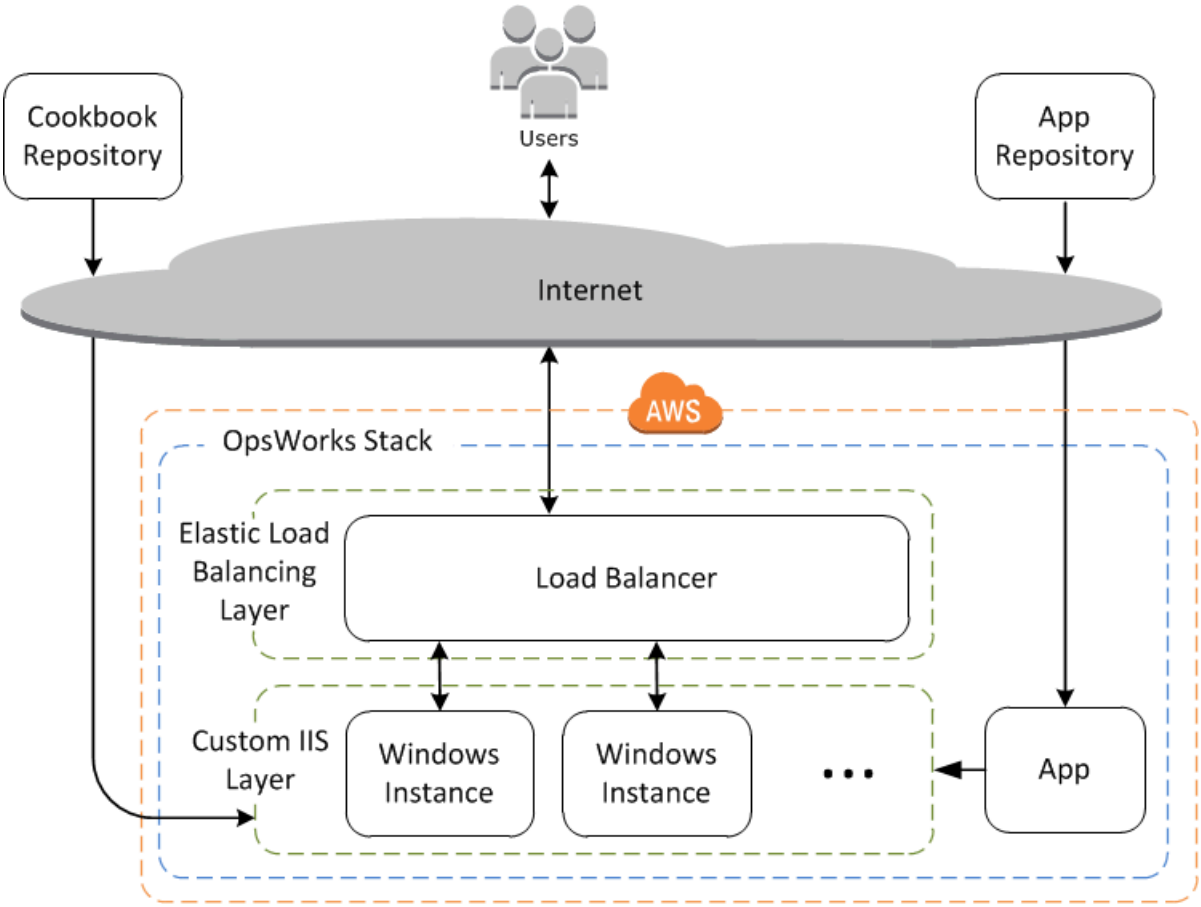
Ora l'app è installata. Puoi aprirla scegliendo Instances (Istanze) nel riquadro Navigation (Navigazione) e scegliendo l'indirizzo IP pubblico dell'istanza. Questa operazione invia una richiesta HTTP all'istanza e dovresti visualizzare nel tuo browser un elemento come quello che segue.

Hello World!

Fase 3: scalabilità orizzontale IISExample

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se le richieste utente in entrata cominciano ad avvicinarsi al limite di ciò che puoi gestire con una singola istanza t2.micro, dovrai incrementare la capacità del tuo server. Puoi passare a un'istanza di dimensioni maggiori, ma questa dispone di limiti. Un approccio più flessibile riguarda l'aggiunta di istanze al tuo stack e il loro inserimento dietro un sistema di bilanciamento del carico. L'architettura di base sarà simile all'esempio seguente.



Tra gli altri vantaggi, questo approccio è molto più efficace di una singola istanza large.

- Se una delle istanze ha esito negativo, il sistema di bilanciamento del carico distribuirà le richieste in entrata per le altre istanze, e la tua applicazione continuerà a funzionare.
- Se poni le istanze in diverse zone di disponibilità (pratica consigliata), la tua applicazione continuerà a funzionare anche se una zona di disponibilità rileva problemi.

AWS OpsWorks Stacks semplifica la scalabilità orizzontale degli stack. Questa sezione descrive le basi su come scalare uno stack aggiungendo una seconda istanza di PHP App Server 24 ore su 24, 7 giorni su 7 IISExample e inserendo entrambe le istanze dietro un load balancer Elastic Load Balancing. È possibile estendere facilmente la procedura per aggiungere un numero arbitrario di istanze 24 ore su 24, 7 giorni su 7, oppure utilizzare istanze basate sul tempo per consentire a Stacks di ridimensionare automaticamente lo stack. AWS OpsWorks Per ulteriori informazioni, consulta Gestione del carico con istanze basate sul tempo e sul carico.

Aggiungere un sistema di bilanciamento del carico

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Elastic Load Balancing è un servizio AWS che distribuisce automaticamente il traffico delle applicazioni in entrata su più istanze Amazon. EC2 Un sistema di bilanciamento del carico è in grado di elaborare due funzioni. La prima è certamente quella di equalizzare il carico di lavoro dei server dell'applicazione. Molti siti preferiscono isolare i rispettivi server dell'applicazione e i database dall'accesso diretto degli utenti. Oltre a distribuire il traffico, Elastic Load Balancing esegue le seguenti operazioni:

- Rileva istanze Amazon EC2 non integre.

  Re-instrada il traffico verso le istanze integre rimanenti, finché quelle non integre non vengono ripristinate.
- Ridimensiona automaticamente la capacità di gestione delle richieste in risposta al traffico in entrata.

> **Note**
>
> AWS OpsWorks Stacks non supporta Application Load Balancer. Puoi usare Classic Load Balancer solo con AWS OpsWorks Stacks.

Sebbene Elastic Load Balancing venga spesso definito un livello, funziona in modo leggermente diverso rispetto agli altri livelli integrati. Invece di creare un layer e aggiungervi istanze, create un load balancer Elastic Load Balancing utilizzando la console EC2 Amazon e poi collegatelo a uno dei livelli esistenti, di solito un livello di application server. AWS OpsWorks Stacks registra quindi le istanze esistenti del layer con il servizio e aggiunge automaticamente tutte le nuove istanze. La procedura seguente illustra come aggiungere un sistema di bilanciamento del carico.

Per collegare un sistema di bilanciamento del carico a un livello IIS personalizzato

1. Usa la EC2 console Amazon per creare un nuovo sistema di bilanciamento del carico per IISExample. Per ulteriori informazioni, consulta Nozioni di base su Elastic Load Balancing. Quando si esegue la procedura guidata Create Load Balancer (Crea sistema di bilanciamento del carico), configurare il sistema di bilanciamento del carico come segue:

   1: Define Load Balancer (Definisci sistema di bilanciamento del carico)

   Assegna al load balancer un nome facilmente riconoscibile, ad esempio IIS-LB, per facilitarne l'individuazione nella console Stacks. AWS OpsWorks Accetta i valori predefiniti per le impostazioni restanti, quindi scegli Next: Assign Security Groups (Successivo: Assegnare i gruppi di sicurezza).

   2: Assign Security Groups (Assegna i gruppi di sicurezza)

   Se il tuo account supporta una VPC predefinita, la procedura guidata visualizza questa pagina per determinare il gruppo di sicurezza del sistema di bilanciamento del carico. Non visualizza questa pagina per Classic. EC2

   Per questa procedura guidata, specifica il default VPC security group (gruppo di sicurezza specifico della VPC), quindi scegli Next: Configure Security Settings (Successivo: configura impostazioni di sicurezza).

3: Configure Security Settings (Configura impostazioni di sicurezza)

Per questa procedura guidata non è necessario che il tuo sistema di bilanciamento del carico utilizzi un listener protetto (ovvero HTTPS o SSL sulle connessioni front-end), quindi scegli Next: Configure Health Check (Successivo: configura controllo dello stato) per continuare.

4: Configure Health Check (Configura controllo dello stato)

Impostare il percorso ping su **/**. Accettate le impostazioni predefinite per le altre impostazioni, quindi scegliete Avanti: Aggiungi EC2 istanze.

5: Aggiungi istanze EC2

AWS OpsWorks Stacks si occupa automaticamente della registrazione delle istanze con il load balancer. Scegli Next Add Tags (Successiva aggiunta di tag) per continuare.

6: Add Tags (Aggiungi tag)

Non utilizzerai tag per questo esempio. Scegli Review and Create (Analizza e crea).

7: Review (Analizza)

Analizza le tue scelte e scegli Create (Crea), quindi Close (Chiudi), azione che lancia il sistema di bilanciamento del carico.

2. Se il tuo account supporta una VPC predefinita, dopo aver lanciato il sistema di bilanciamento del carico devi assicurarti che il suo gruppo di sicurezza abbia appropriate regole di ingresso. La regola predefinita non accetta alcun traffico in ingresso.

   1. Scegli Security Groups nel pannello di EC2 navigazione di Amazon.
   2. Scegli default VPC security group (gruppo di sicurezza predefinito della VPC)
   3. Nella scheda Inbound (In entrata), selezionare Edito (Modifica).
   4. Per questo scenario, impostare Source (Origine) su Anywhere (Ovunque) per indicare al sistema di bilanciamento del carico di accettare il traffico in entrata da qualsiasi indirizzo IP.
   5. Fai clic su Save (Salva).

3. Torna alla console AWS OpsWorks Stacks. Nella pagina Layers (Livelli), scegli Network (Rete).
4. In Elastic Load Balancing (Bilanciamento del carico elastico), seleziona il sistema di bilanciamento del carico IIS-LB creato nella Fase 1 e fai clic su Save (Salva).

Dopo aver collegato il load balancer al layer, AWS OpsWorks Stacks registra automaticamente le istanze correnti del layer e ne aggiunge di nuove non appena sono online.

5. Nella pagina Layers (Livelli), fare clic sul nome del sistema di bilanciamento del carico per aprire la pagina dei dettagli. Un segno di spunta verde accanto all'istanza sulla pagina del sistema di bilanciamento del carico indicherà che l'istanza ha superato un controllo dello stato.

Ora puoi eseguire l'operazione IIS-Example-App inviando una richiesta al sistema di bilanciamento del carico.

Per eseguire IIS-Example-App il bilanciamento del carico

1. Scegli Layers (Livelli). Il sistema di bilanciamento del carico IIS-ELB deve essere elencato come livello e la colonna dello stato deve disporre di un'istanza in verde, la quale indica un'istanza integra.
2. Scegli il nome DNS del load balancer da eseguire. IIS-Example-App Deve essere elencata sotto al nome del sistema di bilanciamento del carico e avere un aspetto simile a `IIS-LB-1802910859.us-west-2.elb.amazonaws.com`. Il sistema di bilanciamento del carico inoltra la richiesta all'istanza e restituisce la risposta, che dovrebbe avere esattamente lo stesso aspetto della risposta che si ottiene quando si fa clic sull'indirizzo IP pubblico dell'istanza.

Hai a disposizione solo un'istanza a questo punto, perciò il sistema di bilanciamento del carico non aggiunge molto. Tuttavia, ora puoi aggiungere altre istanze per il livello.

Per aggiungere un'istanza al livello

1. Scegli Instances (Istanze) e + instance (+ istanza) per aggiungere un'altra istanza al livello.
2. Avviare l'istanza.

Poiché si tratta di istanze nuove, AWS OpsWorks Stacks installa automaticamente gli attuali libri di cucina personalizzati e distribuisce la versione corrente dell'app durante la configurazione. Quando l'istanza è online, AWS OpsWorks Stacks la aggiunge automaticamente al sistema di bilanciamento del carico, in modo che l'istanza inizi immediatamente a gestire le richieste. Per verificare che l'applicazione stia ancora funzionando, puoi scegliere nuovamente il nome DNS del sistema di bilanciamento del carico.

Fasi successive

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

In questa procedura guidata sono state descritte le nozioni di base relative alla configurazione di un semplice stack del server dell'applicazione di Windows. Di seguito sono elencati alcuni suggerimenti su cosa fare in seguito.

- Se desideri saperne di più, Nozioni di base: libri di ricette fornisce un tutorial introduttivo all'implementazione dei libri di cucina e include una serie di AWS OpsWorks esempi specifici per Stacks.
- Puoi aggiungere un layer Amazon Relational Database Service (Amazon RDS) allo stack da utilizzare come server di database backend. Per ulteriori informazioni su come connettere la tua applicazione a un database, consulta Utilizzo di una ricetta personalizzata.

Guida introduttiva ai libri di cucina in Stacks AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Uno stack AWS OpsWorks Stacks a livello di produzione richiede in genere alcune personalizzazioni, il che spesso significa implementare un ricettario Chef personalizzato. Un libro di ricette è un file di tipo pacchetto contenente le informazioni di configurazione, comprese istruzioni chiamate ricette. Una ricetta è un set di una o più istruzioni, scritte con la sintassi del linguaggio Ruby, che specificano le

risorse da utilizzare e l'ordine in cui tali risorse vengono applicate. Una risorsa, come viene utilizzata in Chef, è un'istruzione di una policy di configurazione. Questa procedura dettagliata fornisce un'introduzione di base all'implementazione dei libri di cucina Chef per Stacks. AWS OpsWorks Per ulteriori informazioni su Chef, libri di ricette, ricette e risorse, consulta i collegamenti nella sezione Fasi successive.

Questa procedura guidata descrive principalmente come creare libri di ricette personalizzati. Puoi anche utilizzare i libri di ricette disponibili nella community nei siti Web, come ad esempio i Chef Supermarket. Per aiutarti a iniziare a usare i libri di ricette disponibili nella community, abbiamo incluso le istruzioni per l'utilizzo di un libro di ricette della community disponibile nel supermercato Chef più avanti nella procedura guidata.

Prima di iniziare questa procedura guidata, completa alcune operazioni di configurazione. Se hai già completato una delle altre procedure guidate in questo capitolo, ad esempio Nozioni di base: esempio, hai già soddisfatto i prerequisiti di questa procedura guidata e pertanto puoi avviarla subito. In caso contrario, assicurati di essere conforme a tutti i prerequisiti prima di eseguire questa procedura guidata.

Argomenti

- Fase 1. Creazione del libro di ricette
- Fase 2. Creazione dello stack e dei relativi componenti
- Fase 3. Esecuzione e test della ricetta
- Fase 4. Aggiornamento del libro di ricette per l'installazione di un pacchetto
- Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta
- Fase 6. Aggiornamento del libro di ricette per l'aggiunta di un utente
- Fase 7. Aggiornamento del libro di ricette per la creazione di una directory
- Fase 8. Aggiornamento del libro di ricette per la creazione e la copia di file
- Fase 9. Aggiornamento del libro di ricette per l'esecuzione di un comando
- Fase 10. Aggiornamento del libro di ricette per l'esecuzione di uno script
- Fase 11. Aggiornamento del libro di ricette per la gestione di un servizio
- Fase 12. Aggiornamento del libro di ricette per l'utilizzo di JSON personalizzato
- Fase 13. Aggiornamento del libro di ricette per l'utilizzo di contenitori di dati
- Fase 14. Aggiornamento del libro di ricette per l'utilizzo dell'iterazione
- Fase 15. Aggiornamento del libro di ricette per l'utilizzo della logica condizionale

- Fase 16. Aggiornamento del libro di ricette per l'utilizzo dei libri di ricette della community
- Fase 17: pulizia (opzionale)
- Fasi successive

Fase 1. Creazione del libro di ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Inizia a creare un libro di ricette. Questo libro di ricette non serve per iniziare, ma come base per il resto di questa procedura dettagliata.

> **Note**
>
> Questa fase illustra come creare un libro di ricette manualmente. Puoi velocizzare la creazione di un libro di ricette con il kit di sviluppo Chef (Chef DK) eseguendo il comando chef generate cookbook sulla workstation locale. Tuttavia, questo comando crea più cartelle e file che non sono necessario per questa procedura guidata.

Per creare il libro di ricette

1. Nella workstation locale, creare una directory denominata `opsworks_cookbook_demo`. È possibile utilizzare un nome diverso, ma è necessario assicurarsi di sostituirlo a `opsworks_cookbook_demo` durante l'esecuzione di questa procedura guidata.
2. Nella directory `opsworks_cookbook_demo`, creare un file denominato `metadata.rb` utilizzando un editor di testo. Aggiungere il seguente codice per specificare il nome del libro di ricette. Per ulteriori informazioni su `metadata.rb`, consultare metadata.rb sul sito Web Chef.

   ```
   name "opsworks_cookbook_demo"
   ```

3. Nella directory `opsworks_cookbook_demo` creare una sottodirectory denominata `recipes`. Questa sottodirectory contiene tutte le ricette create per il libro di ricette utilizzando di questa procedura guidata.
4. Nella directory `recipes`, creare un file denominato `default.rb`. Questo file contiene una ricetta con lo stesso nome del file, ma senza la relativa estensione `default`. Aggiungere la seguente riga di codice al file `default.rb`. Questo codice è una ricetta di una sola riga che visualizza un semplice messaggio nel log quando la ricetta viene eseguita:

   ```
   Chef::Log.info("********** Hello, World! **********")
   ```

5. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare un file denominato `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto. Per esempio:

   ```
   tar -czvf opsworks_cookbook_demo.tar.gz opsworks_cookbook_demo/
   ```

   È possibile utilizzare un nome di file diverso, ma è necessario assicurarsi di sostituirlo a `opsworks_cookbook_demo.tar.gz` durante l'esecuzione di questa procedura guidata.

   > **Note**
   >
   > Quando si crea il file `tar` in Windows, la directory superiore deve essere la directory padre del libro di ricette. Questa procedura guidata è stata testata in Linux con il comando tar disponibile nel pacchetto `tar` e su Windows con il comando tar disponibile in Git Bash. L'utilizzo di altri comandi o programmi per la creazione di un file compresso TAR (.tar.gz) potrebbe non funzionare correttamente.

6. Creare un bucket S3 o sceglierne uno esistente. Per ulteriori informazioni, consulta la sezione relativa alla creazione di un bucket.
7. Caricare il file `opsworks_cookbook_demo.tar.gz` nel bucket S3. Per ulteriori informazioni, consulta Aggiungere un oggetto a un bucket.

A questo punto disporrai di un libro di ricette che potrai utilizzare in questa procedura guidata.

Nella fase successiva, crei uno AWS OpsWorks stack Stacks che utilizzerai in seguito per caricare il tuo libro di cucina e per eseguire le ricette del ricettario.

Fase 2. Creazione dello stack e dei relativi componenti

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Crea uno AWS OpsWorks stack Stacks e i relativi componenti, che includono un livello e un'istanza. Nelle fasi successive carichi il libro di ricette nell'istanza e quindi esegui le ricette in esso contenute in tale istanza.

Per creare lo stack

1. Accedi alla console AWS OpsWorks Stacks all'indirizzo /opsworks. https://console.aws.amazon.com
2. Eseguire una delle seguenti operazioni, se possibile:
   - Se viene visualizzata la pagina Welcome to AWS OpsWorks Stacks, scegli Aggiungi il tuo primo stack o Aggiungi il tuo primo stack AWS OpsWorks Stacks (entrambe le scelte hanno la stessa funzione). Viene visualizzata la pagina Add stack (Aggiungi stack).
   - Se viene visualizzata la pagina OpsWorks Dashboard, scegli Aggiungi stack. Viene visualizzata la pagina Add Stack (Aggiungi stack).
3. Scegliere Chef 12 stack (Stack Chef 12).
4. Nella casella Stack name (Nome stack) digitare il nome dello stack, ad esempio **`MyCookbooksDemoStack`**. È possibile utilizzare un nome diverso, ma è necessario assicurarsi di sostituirlo a `MyCookbooksDemoStack` durante l'esecuzione di questa procedura guidata.
5. Per Regione, scegli Stati Uniti occidentali (Oregon).
6. In VPC, eseguire una delle seguenti operazioni:
   - Se è disponibile un VPC, sceglierlo. Per ulteriori informazioni, consulta Esecuzione di uno stack in un VPC.
   - In caso contrario, scegliere No VPC (Nessun VPC).
7. In Use custom Chef cookbooks (Utilizza i libri di ricette di Chef personalizzati), scegliere Yes (Sì).

8. In Repository type (Tipo di repository), scegliere S3 Archive (Archivio S3).

   > **Note**
   >
   > Nella procedura guidata Nozioni di base: Linux è stato scelto Http Archive (Archivio HTTP). Qui assicurarsi di scegliere invece S3 Archive (Archivio S3).

9. Per Repository URL (URL repository), digitare il percorso del file `opsworks_cookbook_demo.tar.gz` in S3. Per ottenere il percorso, nella console S3 selezionare il file `opsworks_cookbook_demo.tar.gz`. Nel riquadro Properties (Proprietà), copiare il valore del campo Link (Collegamento). Dovrebbe essere simile a quanto segue: `https://s3.amazonaws.com/amzn-s3-demo-bucket/opsworks_cookbook_demo.tar.gz`.
10. Se il tuo bucket S3 è privato, che è l'impostazione predefinita, per ID chiave di accesso e chiave di accesso segreta, digita l'ID della chiave di accesso e la chiave di accesso segreta dell'utente IAM che stai utilizzando per questa procedura dettagliata. Per ulteriori informazioni, consulta la sezione relativa alla modifica delle autorizzazioni degli oggetti e quella relativa alla condivisione di un oggetto con altri utenti.
11. Non modificare le impostazioni predefinite per le seguenti opzioni:
    - Default Availability Zone (Zona di disponibilità predefinita) (us-west-2a)
    - Sistema operativo predefinito (Linux e Amazon Linux 2016.09)
    - Default SSH key (Chiave SSH predefinita) (Do not use a default SSH key (Non utilizzare una chiave SSH predefinita))
    - Stack color (Colore stack) (blu scuro)
12. Scegli Advanced (Avanzato).
13. In IAM role (Ruolo IAM), eseguire una delle seguenti operazioni:
    - Se aws-opsworks-service-roledisponibile, sceglilo.
    - Se non aws-opsworks-service-roleè disponibile, scegli Nuovo ruolo IAM.
14. Per il profilo dell'istanza IAM predefinito, procedi in uno dei seguenti modi:
    - Se sono disponibili aws-opsworks-ec2 ruoli, sceglili.
    - Se aws-opsworks-ecil doppio ruolo non è disponibile, scegli Nuovo profilo di istanza IAM.
15. Non modificare le impostazioni predefinite per le seguenti opzioni:

- Default root device type (Tipo di dispositivo radice predefinito) (EBS backed (Supportato da EBS))
- Hostname theme (Tema nome host) (Layer Dependent (Dipendente dal livello))
- OpsWorks Versione dell'agente (versione più recente)
- Custom Chef JSON (JSON Chef personalizzato) (vuoto)
- Sicurezza, utilizza gruppi OpsWorks di sicurezza (Sì)

16. Scegli Aggiungi pila. AWS OpsWorks Stacks crea lo stack e visualizza la pagina. MyCookbooksDemoStack

Per creare un livello

1. Nel riquadro di navigazione del servizio, scegliere Layers (Livelli). Viene visualizzata la pagina Layers (Livelli).
2. Scegliere Add a layer (Aggiungi un livello).
3. Nella OpsWorksscheda, per Nome, digitate. **`MyCookbooksDemoLayer`** È possibile utilizzare un nome diverso, ma è necessario assicurarsi di sostituirlo a `MyCookbooksDemoLayer` durante l'esecuzione di questa procedura guidata.
4. Per Short name (Nome breve), digitare **`cookbooks-demo`**. È possibile utilizzare un nome diverso, ma è necessario assicurarsi di sostituirlo a `cookbooks-demo` durante l'esecuzione di questa procedura guidata.
5. Scegli Aggiungi livello. AWS OpsWorks Stacks aggiunge il livello e visualizza la pagina Livelli.

Per creare e avviare l'istanza

1. Nel riquadro di navigazione del servizio, scegliere Instances (Istanze). Viene visualizzata la pagina Instances (Istanze).
2. Scegli Aggiungi un'istanza.
3. Nella scheda New (Nuovo), scegliere Advanced (Avanzate).
4. Non modificare le impostazioni predefinite per le seguenti opzioni:

   - Hostname (Nome host) (cookbooks-demo1)
   - Size (Dimensione) (c3.large)
   - Sottorete (us-west-2a) ***`IP address`***

- Scaling type (Tipo di dimensionamento) (24/7)
- SSH key (Chiave SSH) (Do not use a default SSH key (Non utilizzare una chiave SSH predefinita))
- Sistema operativo (Amazon Linux 2016.09)
- OpsWorks Versione dell'agente (eredita dallo stack)
- Tenancy (Default - Rely on VPC settings (Predefinita - Basata su impostazioni VPC))
- Root device type (Tipo di dispositivo radice) (EBS backed (Supportato da EBS))
- Volume type (Tipo di volume) (General Purpose (SSD) (Per scopi generici (SSD))
- Volume size (Dimensioni volume) (8)

5. Scegliere Add instance (Aggiungi istanza).
6. Per MyCookbooksDemoLayer, per cookbooks-demo1, per Actions, scegli start. Proseguire solo dopo che in Status (Stato) viene visualizzato online. Questo processo potrebbe richiedere alcuni minuti.

A questo punto si dispone di uno stack, un livello e un'istanza contenente il libro di ricette automaticamente copiato dal bucket S3. Nella fase successiva eseguirai la ricetta predefinita e ne eseguirai il test dall'interno del libro di ricette nell'istanza.

Fase 3. Esecuzione e test della ricetta

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Esegui e prova la `default` ricetta dall'interno del ricettario che AWS OpsWorks Stacks ha copiato nell'istanza. Come già detto in precedenza, si tratta di una ricetta di una sola riga che visualizza un semplice messaggio nel log quando la ricetta viene eseguita.

Per eseguire la ricetta

1. Nel riquadro di navigazione del servizio, scegliere Stack (Stack). Viene visualizzata la pagina MyCookbooksDemoStack.
2. Scegliere Run Command. Viene visualizzata la pagina Run Command.
3. In Command (Comando), scegliere Execute Recipes (Esegui ricette).
4. Per Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::default`**.

   `opsworks_cookbook_demo` è il nome del libro di ricette definito nel file `metadata.rb`. **`default`** è il nome della ricetta da eseguire, ovvero il nome del file `default.rb` nella sottodirectory `recipes` del libro di ricette, senza la relativa estensione.
5. Non modificare le impostazioni predefinite per le seguenti opzioni:

   - Comment (Commento) (vuoto)
   - Advanced (Avanzate), Custom Chef JSON (JSON Chef personalizzato) (vuoto)
   - Istanze (Seleziona tutto selezionato, selezionato, cookbooks-demo1 MyCookbooksDemoLayerselezionato)
6. Scegliere Execute Recipes (Esegui ricette). Viene visualizzata la pagina di esecuzione del comando execute_recipes. Proseguire solo dopo che in Status (Stato) viene visualizzato successful (operazione riuscita). Questo processo potrebbe richiedere alcuni minuti.

Per verificare i risultati della ricetta

1. Con la pagina di esecuzione del comando execute_recipes visualizzata, in cookbooks-demo1, in Log, scegliere show (mostra). Viene visualizzata la pagina del log del comando execute_recipes.
2. Scorrere il log verso il basso e trovare una voce simile alla seguente:

   ```
   [2015-11-13T19:14:39+00:00] INFO: ********** Hello, World! **********
   ```

Hai eseguito la tua prima ricetta correttamente. Nella fase successiva aggiornerai il libro di ricette aggiungendo una ricetta che installa un pacchetto sull'istanza.

Fase 4. Aggiornamento del libro di ricette per l'installazione di un pacchetto

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi aggiornare il libro di ricette aggiungendo una ricetta che installa sull'istanza un pacchetto contenente l'editor di testo GNU Emacs.

Sebbene sia possibile accedere all'istanza con la stessa facilità e installare il pacchetto una sola volta, scrivere una ricetta consente di eseguire la ricetta da AWS OpsWorks Stacks una volta per installare più pacchetti su più istanze in uno stack contemporaneamente.

Per aggiornare il libro di ricette per l'installazione di un pacchetto

1. Tornare alla workstation locale e nella sottodirectory `recipes` della directory `opsworks_cookbook_demo` creare un file denominato `install_package.rb` con il seguente codice:

   ```
   package "Install Emacs" do
     package_name "emacs"
   end
   ```

   Questa ricetta installa il pacchetto `emacs` nell'istanza. Per ulteriori informazioni, passare a package.

   > **Note**
   >
   > Alla ricetta è possibile assegnare qualsiasi nome di file desiderato. Assicurati solo di specificare il nome corretto della ricetta ogni volta che vuoi che AWS OpsWorks Stacks esegua la ricetta.

2. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.

Questa nuova ricetta viene eseguita quando si aggiorna il libro di ricette nell'istanza e quindi si esegue la nuova ricetta dall'interno del libro di ricette aggiornato. La fase successiva descrive questa procedura.

Dopo aver completato il passaggio successivo, sarà possibile eseguire l'accesso all'istanza e quindi digitare emacs nel prompt dei comandi per avviare GNU Emacs. Per ulteriori informazioni, consulta Connessione all'istanza Linux. Per uscire da GNU Emacs, premi Ctrl+X, quindi Ctrl+C.

> **Important**
>
> Per accedere all'istanza, devi prima fornire a AWS OpsWorks Stacks le informazioni sulla tua chiave SSH pubblica (che puoi creare con strumenti come ssh-keygen o PuTTYgen), quindi devi impostare le autorizzazioni sullo `MyCookbooksDemoStack` stack per consentire all'utente di accedere all'istanza. Per istruzioni, consulta Registrazione della chiave SSH pubblica di un utente e Accesso con SSH.

Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi aggiornare il libro di ricette nell'istanza e quindi eseguire la ricetta dall'interno del libro di ricette aggiornato nell'istanza. Nel resto di questa procedura guidata dovrai ripetere questa fase ogni volta che aggiorni il libro di ricette mediante l'aggiunta di una nuova ricetta.

Per aggiornare il libro di ricette nell'istanza

1. Nel riquadro di navigazione del servizio, scegliere Stack (Stack). Viene visualizzata la pagina MyCookbooksDemoStack.
2. Scegliere Run Command. Viene visualizzata la pagina Run Command.
3. In Command (Comando), scegliere Update Custom Cookbooks (Aggiorna libri di ricette personalizzati).
4. Non modificare le impostazioni predefinite per le seguenti opzioni:
   - Comment (Commento) (vuoto)
   - Advanced (Avanzate), Custom Chef JSON (JSON Chef personalizzato) (vuoto)
   - Avanzate, Istanze (Seleziona tutto selezionato, selezionato, MyCookbooksDemoLayercookbooks-demo1 selezionato)
5. Scegli Update Custom Cookbooks (Aggiorna libri di ricette personalizzati). Viene visualizzata la pagina di esecuzione del comando update_custom_cookbooks. Proseguire solo dopo che in Status (Stato) viene visualizzato successful (operazione riuscita). Questo processo potrebbe richiedere alcuni minuti.

Per eseguire la ricetta

1. Nel riquadro di navigazione del servizio, scegliere Stack (Stack). Viene visualizzata la pagina MyCookbooksDemoStack.
2. Scegliere Run Command. Viene visualizzata la pagina Run Command.
3. In Command (Comando), scegliere Execute Recipes (Esegui ricette).
4. In Recipes to execute (Ricette da eseguire), digitare il nome della ricetta da eseguire. La prima volta che si effettua questa operazione, la ricetta è denominata **`opsworks_cookbook_demo::install_package`**.

   > **Note**
   >
   > Quando questa procedura verrà ripetuta più avanti, digitare il nome del libro di ricette (**`opsworks_cookbook_demo`**), seguito da due caratteri di due punti (**`::`**) e dal nome della ricetta senza l'estensione di file `.rb`).

5. Non modificare le impostazioni predefinite per le seguenti opzioni:

- Comment (Commento) (vuoto)
- Advanced (Avanzate), Custom Chef JSON (JSON Chef personalizzato) (vuoto)
- Istanze (Seleziona tutto selezionato, selezionato, cookbooks-demo1 selezionato) MyCookbooksDemoLayer

6. Scegliere Execute Recipes (Esegui ricette). Viene visualizzata la pagina di esecuzione del comando execute_recipes. Proseguire solo dopo che in Status (Stato) viene visualizzato successful (operazione riuscita). Questo processo potrebbe richiedere alcuni minuti.

> **Note**
>
> Non è necessario eseguire manualmente le ricette. È possibile assegnare ricette agli eventi del ciclo di vita di un layer, come gli eventi Setup e Configure, e AWS OpsWorks Stacks eseguirà tali ricette automaticamente quando si verifica l'evento. Per ulteriori informazioni, consulta AWS OpsWorks Eventi del ciclo di vita degli stack.

Nella fase successiva aggiornerai il libro di ricette per aggiungere un utente all'istanza.

Fase 6. Aggiornamento del libro di ricette per l'aggiunta di un utente

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi aggiornare il libro di ricette mediante l'aggiunta di una ricetta che aggiunge un utente locale all'istanza e imposta la home directory e la shell dell'utente. L'operazione è analoga all'esecuzione del comando Linux adduser o useradd o del comando Windows net user. Puoi aggiungere un utente locale a un'istanza, ad esempio, quando desideri controllare l'accesso ai file e alle directory dell'istanza.

Puoi anche gestire gli utenti senza l'utilizzo dei libri di ricette. Per ulteriori informazioni, consulta Gestione degli utenti.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella workstation locale, nella sottodirectory `recipes` della directory `opsworks_cookbook_demo`, creare un file denominato `add_user.rb` con il seguente codice (per ulteriori informazioni, passare a user):

   ```
   user "Add a user" do
     home "/home/jdoe"
     shell "/bin/bash"
     username "jdoe"
   end
   ```

2. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.
4. Seguire la procedura descritta nella Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::add_user`**.

Per testare la ricetta

1. Accedere all'istanza, se non è già stato fatto.
2. Nel prompt dei comandi eseguire il comando seguente per confermare l'aggiunta del nuovo utente:

   ```
   grep jdoe /etc/passwd
   ```

   Vengono visualizzate informazioni sull'utente, inclusi dettagli quali il nome dell'utente, il numero di ID, il numero dell'ID del gruppo, la home directory e la shell, simili a quelle riportate di seguito:

   ```
   jdoe:x:501:502::/home/jdoe:/bin/bash
   ```

Nella fase successiva aggiornerai il libro di ricette per creare una directory nell'istanza.

Fase 7. Aggiornamento del libro di ricette per la creazione di una directory

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi aggiornare il libro di ricette mediante l'aggiunta di una ricetta che aggiunge una directory all'istanza. L'operazione è analoga all'esecuzione del comando Linux mkdir o del comando Windows md omkdir.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella workstation locale, nella sottodirectory `recipes` della directory `opsworks_cookbook_demo` creare un file denominato `create_directory.rb` con il seguente codice: Per ulteriori informazioni, passare a directory.

   ```
   directory "Create a directory" do
     group "root"
     mode "0755"
     owner "ec2-user"
     path "/tmp/create-directory-demo"
   end
   ```

2. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.
4. Seguire la procedura descritta nella Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::create_directory`**.

Per testare la ricetta

1. Accedere all'istanza, se non è già stato fatto.
2. Nel prompt dei comandi eseguire il comando seguente per confermare l'aggiunta della nuova directory:

   ```
   ls -la /tmp/create-directory-demo
   ```

   Vengono visualizzate le informazioni sulla directory appena aggiunta, comprese informazioni quali le autorizzazioni, il nome del proprietario e il nome del gruppo:

   ```
   drwxr-xr-x 2 ec2-user root 4096 Nov 18 00:35 .
   drwxrwxrwt 6 root     root 4096 Nov 24 18:17 ..
   ```

Nella fase successiva aggiornerai il libro di ricette per creare un file nell'istanza.

Fase 8. Aggiornamento del libro di ricette per la creazione e la copia di file

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi aggiornare il libro di ricette mediante l'aggiunta di una ricetta che aggiunge due file all'istanza. Il primo risorsa nella ricetta crea un file completamente con il codice della ricetta. L'operazione è analoga all'esecuzione del comando Linux cat, echo o touch o dei comandi Windows echo ofsutil. Questa tecnica è utile per alcuni file semplici o di piccole dimensioni. La seconda risorsa nella ricetta copia un file nel libro di ricette in un'altra directory nell'istanza. Questa operazione è analoga all'esecuzione del comando Linux cp o del comando Windows copy. Questa tecnica viene utilizzata in presenza di un numero elevato di file complessi o di grandi dimensioni.

Prima di iniziare questa fase, completa la Fase 7. Aggiornamento del libro di ricette per la creazione di una directory per essere sicuro che la directory padre dei file esiste già.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella workstation locale, nella directory `opsworks_cookbook_demo` creare una sottodirectory denominata `files`.
2. Nella sottodirectory `files`, creare un file denominato `hello.txt` con il seguente testo: **`Hello, World!`**
3. Nella sottodirectory `recipes` della directory `opsworks_cookbook_demo`, creare un file denominato `create_files.rb` con il seguente codice. Per ulteriori informazioni, passare a file e cookbook_file.

```
file "Create a file" do
  content "<html>This is a placeholder for the home page.</html>"
  group "root"
  mode "0755"
  owner "ec2-user"
  path "/tmp/create-directory-demo/index.html"
end

cookbook_file "Copy a file" do
  group "root"
  mode "0755"
  owner "ec2-user"
  path "/tmp/create-directory-demo/hello.txt"
  source "hello.txt"
end
```

   La risorsa `file` crea un file nel percorso specificato. La risorsa `cookbook_file` copia il file dalla directory `files` creata nel libro di ricette (Chef prevede di trovare una sottodirectory con nome standard denominata `files` da cui sia possibile copiare i file) in un'altra directory dell'istanza.
4. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
5. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.
6. Seguire la procedura descritta nella Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::create_files`**.

Per testare la ricetta

1. Accedere all'istanza, se non è già stato fatto.
2. Nel prompt dei comandi eseguire i comandi seguenti, una alla volta, per confermare l'aggiunta dei nuovi file:

```
sudo cat /tmp/create-directory-demo/index.html

sudo cat /tmp/create-directory-demo/hello.txt
```

   Viene visualizzato il contenuto dei file:

```
<html>This is a placeholder for the home page.</html>

Hello, World!
```

Nella fase successiva aggiornerai il libro di ricette per eseguire un comando nell'istanza.

Fase 9. Aggiornamento del libro di ricette per l'esecuzione di un comando

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi aggiornare il libro di ricette aggiungendo una ricetta che esegue un comando che crea una chiave SSH per l'istanza.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella workstation locale, nella sottodirectory `recipes` della directory `opsworks_cookbook_demo` creare un file denominato `run_command.rb` con il seguente codice: Per ulteriori informazioni, passare a execute.

```
execute "Create an SSH key" do
  command "ssh-keygen -f /tmp/my-key -N fLyC3jbY"
end
```

2. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.
4. Seguire la procedura descritta nella Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::run_command`**.

Per testare la ricetta

1. Accedere all'istanza, se non è già stato fatto.
2. Nel prompt dei comandi eseguire i comandi seguenti, una alla volta, per confermare la creazione della chiave SSH:

```
sudo cat /tmp/my-key

sudo cat /tmp/my-key.pub
```

Viene visualizzato il contenuto della chiave SSH pubblica e della chiave SSH privata:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,DEF7A09C...541583FA
A5p9dCuo...wp0YYH1c
-----END RSA PRIVATE KEY-----

ssh-rsa AAAAB3N...KaNogZkT root@cookbooks-demo1
```

Nella fase successiva aggiornerai il libro di ricette per eseguire uno script nell'istanza.

Fase 10. Aggiornamento del libro di ricette per l'esecuzione di uno script

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi aggiornare il libro di ricette mediante l'aggiunta di una ricetta che esegue uno script nell'istanza. Questa ricetta crea una directory e quindi crea un file nella directory. La scrittura di una ricetta per eseguire uno script contenente più comandi è più facile che l'esecuzione di tali comandi uno alla volta.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella workstation locale, nella sottodirectory `recipes` della directory `opsworks_cookbook_demo` creare un file denominato `run_script.rb` con il seguente codice: Per ulteriori informazioni, passare a script.

   ```
   script "Run a script" do
     interpreter "bash"
     code <<-EOH
       mkdir -m 777 /tmp/run-script-demo
       touch /tmp/run-script-demo/helloworld.txt
       echo "Hello, World!" > /tmp/run-script-demo/helloworld.txt
     EOH
   end
   ```

2. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.
4. Seguire la procedura descritta nella Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::run_script`**.

Per testare la ricetta

1. Accedere all'istanza, se non è già stato fatto.
2. Nel prompt dei comandi eseguire il comando seguente per confermare l'aggiunta del nuovo file:

   ```
   sudo cat /tmp/run-script-demo/helloworld.txt
   ```

   Viene visualizzato il contenuto dei file:

   ```
   Hello, World!
   ```

Nella fase successiva aggiornerai il libro di ricette per gestire un servizio nell'istanza.

Fase 11. Aggiornamento del libro di ricette per la gestione di un servizio

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi aggiornare il libro di ricette mediante l'aggiunta di una ricetta che gestisce un servizio nell'istanza. L'operazione è analoga all'esecuzione del comando Linux service o del comando Windows net stop, net start e comandi simili. Questa ricetta arresta il servizio crond nell'istanza.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella workstation locale, nella sottodirectory `recipes` della directory `opsworks_cookbook_demo` creare un file denominato `manage_service.rb` con il seguente codice: Per ulteriori informazioni, passare a service.

   ```
   service "Manage a service" do
     action :stop
     service_name "crond"
   end
   ```

2. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.
4. Seguire la procedura descritta nella Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::manage_service`**.

Per testare la ricetta

1. Accedere all'istanza, se non è già stato fatto.
2. Nel prompt dei comandi eseguire il comando seguente per confermare l'arresto del servizio crond:

   ```
   service crond status
   ```

   Viene visualizzato quanto segue:

   ```
   crond is stopped
   ```

3. Per riavviare il servizio crond, eseguire il comando seguente:

   ```
   sudo service crond start
   ```

   Viene visualizzato quanto segue:

   ```
   Starting crond:  [  OK  ]
   ```

4. Per confermare l'avvio del servizio crond, rieseguire il seguente comando:

   ```
   service crond status
   ```

   Verranno visualizzate informazioni simili alle seguenti:

   ```
   crond (pid  3917) is running...
   ```

Nella fase successiva aggiornerai il libro di ricette per fare riferimento alle informazioni archiviate come JSON personalizzato nell'istanza.

Fase 12. Aggiornamento del libro di ricette per l'utilizzo di JSON personalizzato

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi aggiornare il libro di ricette aggiungendo una ricetta che fa riferimento a JSON personalizzato archiviato nell'istanza.

È possibile specificare le informazioni in formato JSON personalizzato ogni volta che è necessario creare, aggiornare o clonare uno stack o quando si esegue un comando relativo a distribuzioni o stack. Ciò risulta utile, ad esempio, per creare una piccola quantità di dati non variabili disponibili per le ricette nell'istanza invece di ottenere tali dati da un database. Per ulteriori informazioni, consulta Utilizzo di un JSON personalizzato.

Per questa procedura guidata utilizzerai un formato JSON personalizzato per fornire alcune informazioni fittizie su una fattura cliente. Il formato JSON personalizzato è descritto più avanti in questa fase.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella workstation locale, nella sottodirectory `recipes` della directory `opsworks_cookbook_demo` creare un file denominato `custom_json.rb` contenente il seguente codice della ricetta:

```
Chef::Log.info("********** For customer '#{node['customer-id']}' invoice
 '#{node['invoice-number']}' **********")
Chef::Log.info("********** Invoice line number 1 is a '#{node['line-items']
['line-1']}' **********")
Chef::Log.info("********** Invoice line number 2 is a '#{node['line-items']
['line-2']}' **********")
```

```
Chef::Log.info("********** Invoice line number 3 is a '#{node['line-items']
['line-3']}' **********")
```

Questa ricetta visualizza i messaggi nel log relativi ai valori nel formato JSON personalizzato.

2. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.
4. Seguire la procedura descritta nella Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::custom_json`**. In Advanced (Avanzate), Custom Chef JSON (JSON Chef personalizzato), digitare il seguente JSON personalizzato:

```
{
  "customer-id": "0123",
  "invoice-number": "9876",
  "line-items": {
    "line-1": "tractor",
    "line-2": "passenger car",
    "line-3": "trailer"
  }
}
```

Per testare la ricetta

1. Con la pagina di esecuzione del comando execute_recipes visualizzata dalle precedenti procedure, in cookbooks-demo1, in Log, scegliere show (mostra). Viene visualizzata la pagina del log del comando execute_recipes.
2. Scorrere il log verso il basso e trovare le voci simili alle seguenti:

```
[2015-11-14T14:18:30+00:00] INFO: ********** For customer '0123' invoice '9876'
 **********
[2015-11-14T14:18:30+00:00] INFO: ********** Invoice line number 1 is a 'tractor'
 **********
[2015-11-14T14:18:30+00:00] INFO: ********** Invoice line number 2 is a 'passenger
 car' **********
```

```
[2015-11-14T14:18:30+00:00] INFO: ********** Invoice line number 3 is a 'trailer'
 **********
```

Queste voci vengono visualizzate in base al JSON personalizzato immesso nella casella Advanced (Avanzate), Custom Chef JSON (JSON Chef personalizzato).

Nella fase successiva, aggiornerai il ricettario per ottenere informazioni dai data bag, che sono raccolte di impostazioni dello stack che AWS OpsWorks Stacks memorizza su ogni istanza.

Fase 13. Aggiornamento del libro di ricette per l'utilizzo di contenitori di dati

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Aggiorna il tuo ricettario aggiungendo una ricetta che faccia riferimento alle impostazioni dello stack che AWS OpsWorks Stacks memorizza sull'istanza in un set di data bag. Questa ricetta mostra i messaggi nel log sulle impostazioni dello stack specifiche archiviate nell'istanza. Per ulteriori informazioni, consulta AWS OpsWorks Riferimento a Stacks Data Bag.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella workstation locale, nella sottodirectory `recipes` della directory `opsworks_cookbook_demo` creare un file denominato `data_bags.rb` contenente il seguente codice:

```
instance = search("aws_opsworks_instance").first
layer = search("aws_opsworks_layer").first
stack = search("aws_opsworks_stack").first

Chef::Log.info("********** This instance's instance ID is
 '#{instance['instance_id']}' **********")
Chef::Log.info("********** This instance's public IP address is
 '#{instance['public_ip']}' **********")
```

```
Chef::Log.info("********** This instance belongs to the layer '#{layer['name']}'
 **********")
Chef::Log.info("********** This instance belongs to the stack '#{stack['name']}'
 **********")
Chef::Log.info("********** This stack gets its cookbooks from
 '#{stack['custom_cookbooks_source']['url']}' **********")
```

Questa ricetta mostra i messaggi nel log sulle impostazioni dello stack specifiche archiviate nell'istanza.

2. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.
4. Seguire la procedura descritta nella Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::data_bags`**.

Per testare la ricetta

1. Con la pagina di esecuzione del comando execute_recipes visualizzata dalla precedente procedura, in cookbooks-demo1, in Log, scegliere show (mostra). Viene visualizzata la pagina del log del comando execute_recipes.
2. Scorrere il log verso il basso e trovare le voci simili alle seguenti:

```
[2015-11-14T14:39:06+00:00] INFO: ********** This instance's instance ID is
 'f80fa119-81ab-4c3c-883d-6028e52c89EX' **********
[2015-11-14T14:39:06+00:00] INFO: ********** This instance's public IP address is
 '192.0.2.0' **********
[2015-11-14T14:39:06+00:00] INFO: ********** This instance belongs to the layer
 'MyCookbooksDemoLayer' **********
[2015-11-14T14:39:06+00:00] INFO: ********** This instance belongs to the stack
 'MyCookbooksDemoStack' **********
[2015-11-14T14:39:06+00:00] INFO: ********** This stack gets its cookbooks from
 'https://s3.amazonaws.com/amzn-s3-demo-bucket/opsworks_cookbook_demo.tar.gz'
 **********
```

Questa ricetta mostra i messaggi sulle impostazioni dello stack specifiche archiviate nell'istanza.

Nella fase successiva aggiornerai il libro di ricette per eseguire il codice della ricetta più volte.

Fase 14. Aggiornamento del libro di ricette per l'utilizzo dell'iterazione

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi aggiornare il libro di ricette aggiungendo una ricetta che utilizza l'iterazione, una tecnica che ripete il codice della ricetta più volte. Questa ricetta visualizza i messaggi nel log per un contenitore di dati contenente più contenuti.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella workstation locale, nella sottodirectory `recipes` della directory `opsworks_cookbook_demo` creare un file denominato `iteration_demo.rb` contenente il seguente codice:

```
stack = search("aws_opsworks_stack").first
Chef::Log.info("********** Content of 'custom_cookbooks_source' **********")

stack["custom_cookbooks_source"].each do |content|
  Chef::Log.info("********** '#{content}' **********")
end
```

> **Note**
>
> La scrittura del codice della ricetta precedente è un'operazione più breve, più flessibile e meno soggetta a errori rispetto alla scrittura del seguente codice della ricetta che non utilizza l'iterazione:
>
> ```
> stack = search("aws_opsworks_stack").first
> Chef::Log.info("********** Content of 'custom_cookbooks_source' **********")
> ```

```
Chef::Log::info("********** '[\"type\", \"#{stack['custom_cookbooks_source']
['type']}\"]' **********")
Chef::Log::info("********** '[\"url\", \"#{stack['custom_cookbooks_source']
['url']}\"]' **********")
Chef::Log::info("********** '[\"username\",
 \"#{stack['custom_cookbooks_source']['username']}\"]' **********")
Chef::Log::info("********** '[\"password\",
 \"#{stack['custom_cookbooks_source']['password']}\"]' **********")
Chef::Log::info("********** '[\"ssh_key\",
 \"#{stack['custom_cookbooks_source']['ssh_key']}\"]' **********")
Chef::Log::info("********** '[\"revision\",
 \"#{stack['custom_cookbooks_source']['revision']}\"]' **********")
```

2. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.
4. Seguire la procedura descritta nella [Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta] per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::iteration_demo`**.

Per testare la ricetta

1. Con la pagina di esecuzione del comando execute_recipes visualizzata dalle precedenti procedure, in cookbooks-demo1, in Log, scegliere show (mostra). Viene visualizzata la pagina del log del comando execute_recipes.
2. Scorrere il log verso il basso e trovare le voci simili alle seguenti:

```
[2015-11-16T19:56:56+00:00] INFO: ********** Content of 'custom_cookbooks_source'
 **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["type", "s3"]' **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["url", "https://s3.amazonaws.com/
amzn-s3-demo-bucket/opsworks_cookbook_demo.tar.gz"]' **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["username", "secret-key-value"]'
 **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["password", "secret-access-key-
value"]' **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["ssh_key", nil]' **********
```

```
[2015-11-16T19:56:56+00:00] INFO: ********** '["revision", nil]' **********
```

Questa ricetta visualizza i messaggi nel log per un contenitore di dati contenente più contenuti. L'elemento del contenitore di dati è incluso nel contenitore di dati `aws_opsworks_stack`. L'elemento del contenitore di dati include un contenuto denominato `custom_cookbooks_source`. All'interno di questo contenuto sono disponibili sei contenuti denominati `type`, `url`, `username`, `password`, `ssh_key`, e `revision`. Vengono visualizzati anche i relativi valori.

Nella fase successiva aggiornerai il libro di ricette per l'esecuzione del codice della ricetta solo se vengono soddisfatte determinate condizioni.

Fase 15. Aggiornamento del libro di ricette per l'utilizzo della logica condizionale

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ora puoi aggiornare il libro di ricette aggiungendo una ricetta che utilizza la logica condizionale, una tecnica che esegue il codice solo se vengono soddisfatte determinate condizioni. Per ulteriori informazioni, consulta le sezioni relative alle istruzioni if e alle istruzioni case.

Questa ricetta esegue due operazioni in base al contenuto dei contenitori di dati: nel log visualizza un messaggio che identifica il sistema operativo in cui l'istanza è in esecuzione e, solo se il sistema operativo è Linux, installa un pacchetto utilizzando il programma di gestione del pacchetto corretto per la distribuzione Linux. Questo pacchetto è denominato struttura; si tratta di una semplice applicazione per la visualizzazione di elenchi di directory.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella workstation locale, nella sottodirectory `recipes` della directory `opsworks_cookbook_demo directory` creare un file denominato `conditional_logic.rb` contenente il seguente codice:

```
instance = search("aws_opsworks_instance").first
os = instance["os"]

if os == "Red Hat Enterprise Linux 7"
  Chef::Log.info("********** Operating system is Red Hat Enterprise Linux.
 **********")
elsif os == "Ubuntu 14.04 LTS" || os == "Ubuntu 16.04 LTS" || os == "Ubuntu 18.04
 LTS"
  Chef::Log.info("********** Operating system is Ubuntu. **********")
elsif os == "Microsoft Windows Server 2012 R2 Base"
  Chef::Log.info("********** Operating system is Windows. **********")
elsif os == "Amazon Linux 2015.03" || os == "Amazon Linux 2015.09" || os == "Amazon
 Linux 2016.03" || os == "Amazon Linux 2016.09" || os == "Amazon Linux 2017.03"
 || os == "Amazon Linux 2017.09" || os == "Amazon Linux 2018.03" || os == "Amazon
 Linux 2"
  Chef::Log.info("********** Operating system is Amazon Linux. **********")
elsif os == "CentOS Linux 7"
  Chef::Log.info("********** Operating system is CentOS 7. **********")
else
  Chef::Log.info("********** Cannot determine operating system. **********")
end

case os
when "Ubuntu 14.04 LTS", "Ubuntu 16.04 LTS", "Ubuntu 18.04 LTS"
  apt_package "Install a package with apt-get" do
    package_name "tree"
  end
when "Amazon Linux 2015.03", "Amazon Linux 2015.09", "Amazon Linux 2016.03",
 "Amazon Linux 2016.09", "Amazon Linux 2017.03", "Amazon Linux 2017.09", "Amazon
 Linux 2018.03", "Amazon Linux 2", "Red Hat Enterprise Linux 7", "CentOS Linux 7"
  yum_package "Install a package with yum" do
    package_name "tree"
  end
else
  Chef::Log.info("********** Cannot determine operating system type, or operating
 system is not Linux. Package not installed. **********")
end
```

2. Al terminale o al prompt dei comandi, utilizzare il comando tar per creare una nuova versione del file `opsworks_cookbook_demo.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e il relativo contenuto aggiornato.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato nel bucket S3.

4. Seguire la procedura descritta nella Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::conditional_logic`**.

Per testare la ricetta

1. Con la pagina di esecuzione del comando execute_recipes visualizzata dalle precedenti procedure, in cookbooks-demo1, in Log, scegliere show (mostra). Viene visualizzata la pagina del log del comando execute_recipes.
2. Scorrere il log verso il basso e trovare una voce simile alla seguente:

   ```
   [2015-11-16T19:59:05+00:00] INFO: ********** Operating system is Amazon Linux.
    **********
   ```

   Poiché il sistema operativo dell'istanza è Amazon Linux 2016.09, nel registro verrà visualizzata solo la voce precedente (delle cinque possibili voci del codice della ricetta).
3. Se il sistema operativo è Linux, la ricetta installa il pacchetto della struttura. Per visualizzare una rappresentazione visiva del contenuto di una directory, digitare **`tree`** al prompt dei comandi dalla directory desiderata o dal percorso della directory desiderata (ad esempio, `tree /var/chef/runs`).

Nella fase successiva aggiornerai il libro di ricette per l'utilizzo di funzionalità da un libro di ricette esterno fornito dalla community Chef.

Fase 16. Aggiornamento del libro di ricette per l'utilizzo dei libri di ricette della community

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi infine aggiornare il libro di ricette per l'utilizzo di funzionalità disponibili in un libro di ricette esterno fornito dalla community Chef. Il libro di ricette esterno utilizzato per questa procedura guidata è disponibile nel supermercato Chef, una posizione tradizionale per accedere ai libri di ricette esterni di Chef. Questo libro di ricette esterno fornisce una risorsa personalizzata che ti permette di scaricare e installare applicazioni, in modo analogo a quanto descritto nella Fase 4. Aggiornamento del libro di ricette per l'installazione di un pacchetto. Tuttavia, questa risorsa può installare applicazioni Web e altri tipi di applicazione in aggiunta ai pacchetti.

Quando un libro di ricette dipende da un altro libro di ricette, devi specificare una dipendenza rispetto all'altro libro di ricette. Per dichiarare e gestire le dipendenze dei libri di ricette, consigliamo di usare uno strumento denominato Berkshelf. Per ulteriori informazioni su come installare Berkshelf nella workstation locale, consulta la pagina relativa alle informazioni su Berkshelf sul sito Web Chef.

Dopo aver installato Berkshelf, esegui queste procedure per dichiarare le dipendenze del libro di ricette e quindi creare una ricetta che richiama la risorsa nel libro di ricette esterno:

Per dichiarare la dipendenza del libro di ricette

1. Nella workstation locale, nella directory `opsworks_cookbook_demo` aggiungere la seguente riga alla fine del file `metadata.rb`:

   ```
   depends "application", "5.0.0"
   ```

   In questo modo viene dichiarata una dipendenza da un libro di ricette denominato `application`, versione 5.0.0.
2. Nella radice della directory `opsworks_cookbook_demo`, eseguire il seguente comando. Il punto alla fine del comando è intenzionale.

   ```
   berks init .
   ```

   Berkshelf crea una serie di cartelle e file che è possibile utilizzare in seguito per altri scenari avanzati. L'unico file necessario per questa procedura guidata è il file denominato `Berksfile`.
3. Aggiungere la seguente riga alla fine del file `Berksfile`:

   ```
   cookbook "application", "5.0.0"
   ```

   In questo modo si indica a Berkshelf che si desidera utilizzare il libro di ricette dell'applicazione versione 5.0.0, da cui Berkshelf può scaricare Chef Supermarket.

4. Al terminale o prompt dei comandi eseguire il comando seguente dalla radice della directory `opsworks_cookbook_demo`:

   ```
   berks install
   ```

   Berkshelf crea un elenco delle dipendenze sia per il libro di ricette personale che per il libro di ricette dell'applicazione. Berkshelf utilizza questo elenco di dipendenze nella procedura successiva.

Per aggiornare il libro di ricette nell'istanza ed eseguire la nuova ricetta

1. Nella sottodirectory `recipes` della directory `opsworks_cookbook_demo`, creare un file denominato `dependencies_demo.rb` contenente il seguente codice:

   ```
   application "Install NetHack" do
     package "nethack.x86_64"
   end
   ```

   Questa ricetta dipende dalla risorsa dell'applicazione contenuta nel ricettario dell'applicazione per installare il popolare gioco di avventura testuale sull'istanza. NetHack È tuttavia possibile utilizzare qualsiasi nome di pacchetto desiderato, a condizione che il pacchetto sia immediatamente disponibile nel programma di gestione del pacchetto sull'istanza.
2. Nella radice della directory `opsworks_cookbook_demo`, eseguire il seguente comando:

   ```
   berks package
   ```

   Berkshelf utilizza l'elenco delle dipendenze dalla procedura precedente per creare un file denominato `cookbooks-`*`timestamp`*`.tar.gz`, che contiene la directory `opsworks_cookbook_demo` e i relativi contenuti aggiornati, inclusi i libri di ricette dipendenti del libro di ricette. Rinominare questo file `opsworks_cookbook_demo.tar.gz`.
3. Caricare il file `opsworks_cookbook_demo.tar.gz` aggiornato e rinominato nel bucket S3.
4. Seguire la procedura descritta nella Fase 5. Aggiornamento del libro di ricette nell'istanza ed esecuzione della ricetta per aggiornare il libro di ricette nell'istanza e per eseguire la ricetta. Nella procedura illustrata nella sezione "Per eseguire la ricetta", in Recipes to execute (Ricette da eseguire), digitare **`opsworks_cookbook_demo::dependencies_demo`**.

5. Dopo aver eseguito la ricetta, dovrebbe essere possibile effettuare l'accesso all'istanza e quindi digitare **`nethack`** al prompt dei comandi per cominciare a giocare. (Per ulteriori informazioni sul gioco, consulta NetHackla Guida.) NetHack

Nel passaggio successivo, puoi ripulire le AWS risorse utilizzate per questa procedura dettagliata. Questa fase è opzionale. Potresti voler continuare a utilizzare queste AWS risorse man mano che continui a saperne di più su AWS OpsWorks Stacks. Tuttavia, il mantenimento di queste AWS risorse potrebbe comportare alcuni addebiti continui sul tuo AWS account. Se desideri conservare queste AWS risorse per un uso successivo, ora hai completato questa procedura dettagliata e puoi passare a. Fasi successive

Fase 17: pulizia (opzionale)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per evitare di incorrere in costi aggiuntivi AWS sul tuo account, puoi eliminare le AWS risorse utilizzate per questa procedura dettagliata. Queste AWS risorse includono il bucket S3, lo AWS OpsWorks stack Stacks e i componenti dello stack. (Per ulteriori informazioni, consulta OpsWorks Prezzi di AWS). Tuttavia, potresti voler continuare a utilizzare queste AWS risorse man mano che continui a saperne di più su AWS OpsWorks Stacks. Se desideri mantenere disponibili queste AWS risorse, ora hai completato questa procedura dettagliata e puoi passare a. Fasi successive

I contenuti archiviati nelle risorse create per questa procedura guidata possono contenere informazioni identificative personali. Se non si desidera che queste informazioni vengano archiviate da AWS, seguire le fasi in questo argomento.

Per eliminare il bucket S3

- Consulta la sezione relativa all'eliminazione del bucket Amazon S3.

Per eliminare l'istanza per lo stack

1. Nella console AWS OpsWorks Stacks, nel riquadro di navigazione del servizio, nel riquadro di navigazione del servizio, scegliere Instances (Istanze). Viene visualizzata la pagina Instances (Istanze).
2. Per MyCookbooksDemoLayer, per cookbooks-demo1, per Actions, scegli stop. Quando viene visualizzato il messaggio di conferma, scegliere Stop (Arresta).
3. Le seguenti modifiche durano diversi minuti. Non procedere finché tutti le procedure seguenti non sono terminate.

   - L'opzione Status (Stato) passa da online a stopping (arresto in corso) e infine a stopped (arrestato).
   - La voce online cambia da 1 in 0.
   - La voce shutting down (chiusura in corso) cambia da 0 in 1 e infine in 0.
   - La voce stopped (arrestato) cambia da 0 in 1.
4. In Actions (Operazioni), scegliere delete (eliminazione). Quando viene visualizzato il messaggio di conferma, scegliere Delete (Elimina). AWS OpsWorks Stacks elimina l'istanza e visualizza Nessuna istanza.

Per eliminare lo stack

1. Nel riquadro di navigazione del servizio, scegliere Stack (Stack). Viene visualizzata la pagina MyCookbooksDemoStack.
2. Scegliere Delete Stack (Elimina stack). Quando viene visualizzato il messaggio di conferma, scegliere Delete (Elimina). AWS OpsWorks Stacks elimina lo stack e visualizza la pagina Dashboard.

Facoltativamente, puoi eliminare l'utente IAM e la coppia di EC2 chiavi Amazon che hai usato per questa procedura dettagliata, se non desideri riutilizzarli per accedere ad altri AWS servizi e istanze. EC2 Per istruzioni, consulta Eliminazione di un utente IAM e di coppie di EC2 chiavi Amazon e istanze Linux.

A questo punto hai completato la procedura guidata. Per ulteriori informazioni, consulta Fasi successive.

Fasi successive

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ora che hai completato questa procedura dettagliata, puoi saperne di più sul supporto di AWS OpsWorks Stacks per i libri di cucina Chef consultando le seguenti risorse:

- Libri di ricette e ricette— Descrive le versioni di Chef e Ruby attualmente supportate da Stacks. AWS OpsWorks Illustra inoltre come installare e aggiornare i libri di ricette personalizzati sulle istanze e come eseguire ricette sulle istanze.
- Learn Chef: fornisce collegamenti ai tutorial di Chef, a una libreria di competenze di Chef, alla documentazione completa di Chef e ai corsi di training di Chef.
- Tutto su Chef: fornisce la documentazione completa di Chef. Tra gli specifici argomenti di interesse è disponibile quanto segue:
  - About Cookbooks (Informazioni sui libri di ricette): descrive i componenti dei principali libri di ricette, ad esempio attributi, ricette, file, metadati e modelli.
  - About Recipes (Informazioni sulle ricette) : descrive i concetti fondamentali relativi alle ricette, ad esempio come utilizzare i contenitori di dati, includere altre ricette e usare il codice Ruby nelle ricette.
  - Resources (Risorse): descrive come utilizzare le risorse predefinite di Chef, ad esempio `apt_package`, `cookbook_file`, `directory`, `execute`, `file` e `package`.
  - About the Recipe DSL (Informazioni sul linguaggio specifico del dominio (DSL) della ricetta) : descrive come scrivere il codice per le ricette di Chef con istruzioni quali, ad esempio, `if`, `case`, `data_bag`, `data_bag_item` e `search`.
- About Templates (Informazioni su modelli): descrive come utilizzare i modelli ERB (Embedded Ruby) per generare dinamicamente i file di testo statico, ad esempio i file di configurazione.
- Learning Tracks: descrive come utilizzare Chef per gestire un'istanza, gestire un'app web di base, sviluppare e testare il codice dell'infrastruttura, utilizzare Chef Analytics e altro ancora.

- Learning Chef: un'introduzione a Chef. Pubblicato da O'Reilly Media.
- Learning Chef code examples (Introduzione al codice di esempio di Chef): fornisce codici di esempio a supporto del manuale Learning Chef pubblicato da O'Reilly Media.

AWS OpsWorks Le migliori pratiche di Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le strategie, le tecniche e i suggerimenti contenuti in questa sezione possono aiutarti a ottenere il massimo beneficio e risultati ottimali da Stacks. AWS OpsWorks

Argomenti

- Best practice: storage del dispositivo root per le istanze
- Best practice: ottimizzazione del numero di server di applicazioni
- Best practice: Gestione delle autorizzazioni
- Best practice: gestione e distribuzione di app e libri di ricette
- Creazione di pacchetti di dipendenze di libri di ricette a livello locale

Best practice: storage del dispositivo root per le istanze

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo argomento non si applica alle istanze Windows, che devono essere supportate da Amazon Elastic Block Store.

Le istanze Amazon Elastic Compute Cloud (Amazon EC2) Linux hanno le seguenti opzioni di storage su dispositivo root.

- Istanze supportate dall'archivio delle istanze: il dispositivo root è temporaneo.

  Se arresti l'istanza, i dati sul dispositivo root vengono rimossi e non possono essere recuperati. Per ulteriori informazioni, consulta Amazon EC2 Instance Store.
- Istanze supportate da Amazon EBS: il dispositivo principale è un volume Amazon EBS.

  Se interrompi l'istanza, il volume Amazon EBS persiste. Se riavvii l'istanza, il volume viene automaticamente rimontata. Questa operazione ripristina lo stato dell'istanza e tutti i dati archiviati. Puoi anche montare il volume su un'altra istanza. Per ulteriori informazioni, consulta la sezione relativa ad Amazon Elastic Block Store (Amazon EBS).

Tieni presente le seguenti considerazioni al momento di scegliere l'opzione relativa allo storage del dispositivo root da utilizzare.

Tempo di avvio

Dopo l'avvio iniziale, le istanze Amazon EBS in genere si riavviano più velocemente.

Il tempo di avvio iniziale è circa lo stesso per entrambi i tipi di storage. Entrambi i tipi devono eseguire una configurazione completa, che prevede attività relativamente dispendiose in termini di tempo, ad esempio l'installazione di pacchetti da archivi remoti. Tuttavia, puoi riscontrare le seguenti distinzioni quando in seguito riavvii un'istanza:

- Le istanze supportate da instance store eseguono le stesse attività di configurazione eseguite durante l'avvio iniziale, inclusa l'installazione dei pacchetti.

  Un riavvio richiede lo stesso tempo dell'avvio iniziale.
- Le istanze Amazon EBS-Back rimontano il volume root ed eseguono le ricette di installazione.

Il riavvio è in genere significativamente più veloce rispetto all'avvio iniziale, perché le ricette di impostazione (Configurazione) non devono eseguire attività quali, ad esempio, la reinstallazione di pacchetti già installati sul volume root.

Costo

Le istanze supportate da Amazon EBS sono più costose:

- Con un'istanza supportata da instance store, paghi solo quando l'istanza è in esecuzione.
- Con le istanze supportate da Amazon EBS, paghi per il volume di Amazon EBS indipendentemente dal fatto che l'istanza sia in esecuzione o meno.

  Per ulteriori informazioni, consulta Prezzi di Amazon EBS.

Registrazione

Le istanze supportate da Amazon EBS conservano automaticamente i log:

- Con le istanze supportate da instance store, i log vengono rimossi quando l'istanza viene arrestata.

  È necessario recuperare i log prima di interrompere l'istanza o utilizzare un servizio come Logs per archiviare i CloudWatch log selezionati in remoto.
- Con un'istanza supportata da Amazon EBS, i log vengono archiviati nel volume Amazon EBS.

  Puoi visualizzarli riavviando l'istanza o montando il volume a un'altra istanza.

Dipendenze

I due tipi di storage dispongono di dipendenze diverse:

- Le istanze supportate da Instance-store dipendono da Amazon S3.

  Quando avvii l'istanza, deve scaricare l'AMI da Amazon S3.
- Le istanze supportate da Amazon EBS dipendono da Amazon EBS.

  Quando avvii l'istanza, deve montare il volume root di Amazon EBS.

Raccomandazione: se non sei sicuro del tipo di storage più adatto alle tue esigenze, ti consigliamo di iniziare con le istanze Amazon EBS. Sebbene si debba sostenere una spesa modesta per i volumi Amazon EBS, il rischio di perdita involontaria dei dati è minore.

Best practice: ottimizzazione del numero di server di applicazioni

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Uno stack di produzione in genere include più server di applicazione distribuiti su più zone di disponibilità. Tuttavia il numero di richieste in entrata può variare notevolmente in base all'ora del giorno o al giorno della settimana. È possibile eseguire solo i server sufficienti a gestire il massimo carico anticipato, ma la maggior parte delle volte si arriva a pagare per più capacità di server rispetto a quella necessaria. Per eseguire il proprio sito in modo efficiente, è prassi consigliata far corrispondere il numero di server all'attuale volume di richieste.

AWS OpsWorks Stacks offre tre modi per gestire il numero di istanze del server.

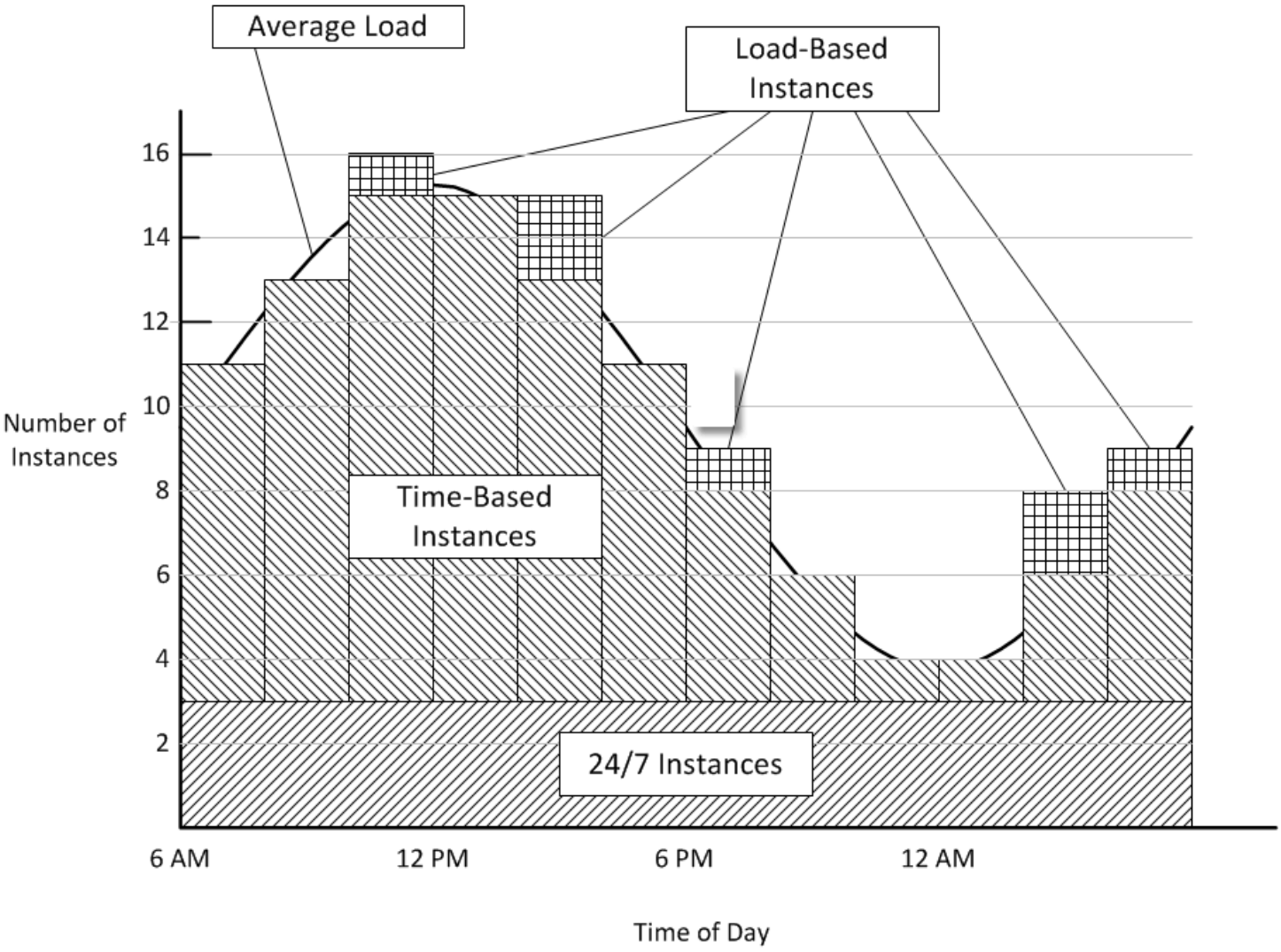
- 24 ore su 24, 7 giorni su 7, le istanze vengono avviate manualmente ed eseguite finché non vengono arrestate manualmente.
- Le istanze basate sul tempo vengono avviate e interrotte automaticamente da AWS OpsWorks Stacks in base a una pianificazione specificata dall'utente.
- Le istanze basate sul caricamento vengono avviate e interrotte automaticamente da AWS OpsWorks Stacks quando superano una soglia per una metrica di carico specificata dall'utente, ad esempio l'utilizzo della CPU o della memoria.

> **Note**
>
> Dopo aver creato e configurato l'orario dello stack e le istanze basate sul carico, AWS OpsWorks Stacks le avvia e le arresta automaticamente in base alla configurazione specificata. Non è necessario toccarle nuovamente a meno che non si decida di modificare la configurazione o il numero di istanze.

Raccomandazione: se si gestiscono gli stack con più di qualche istanza di server di applicazione, è consigliato l'utilizzo di una combinazione di tutti e tre i tipi di istanza. Di seguito è riportato un esempio di come organizzare una capacità di server dello stack per gestire un volume di richieste giornaliero variabile con le seguenti caratteristiche.

- Il volume di richieste medio varia in modo sinusoidale durante la giornata.
- Il volume di richieste medio minimo richiede cinque istanze di server di applicazione.
- Il volume di richieste medio massimo richiede sedici istanze di server di applicazione.
- I picchi del volume di richieste di solito possono essere gestiti da una o due istanze di server di applicazione.

Si tratta di un modello comodo ai fini della discussione, ma è possibile adeguarlo a qualsiasi variazione del volume di richieste e, inoltre, estenderlo per gestire le variazioni settimanali. Il diagramma seguente illustra come utilizzare i tre tipi di istanza per gestire questo volume di richieste.



Questo esempio presenta le caratteristiche seguenti:

- Lo stack ha tre istanze di 24 ore su 24, 7 giorni su 7, che sono sempre attive e gestiscono il carico di base.
- Lo stack ha 12 istanze basato sul tempo, che vengono configurate per gestire la variazione giornaliera media.

  Una viene eseguita dalle 22:00 alle 02:00, altre due vengono eseguite dalle 20:00 alle 22:00 e dalle 02:00 alle 04:00 e così via. Per semplicità, il diagramma modifica il numero di istanze basate sul tempo ogni due ore, ma è possibile modificare il numero ogni ora se si desidera un controllo ancora più accurato.

- La stack ha un numero sufficiente di istanze basate sul carico per gestire i picchi di traffico che superano ciò che può essere gestita dalle istanze 24 ore su 24, 7 giorni su 7 e quelle basate sul tempo.

  AWS OpsWorks Stacks avvia le istanze basate sul caricamento solo quando il carico su tutti i server attualmente in esecuzione supera le metriche specificate. Il costo per le istanze non in esecuzione è minimo (istanze supportate da Amazon EBS) o nullo (istanze supportate dall'archivio delle istanze), quindi la pratica consigliata è crearne una quantità sufficiente per gestire comodamente i volumi di richieste massimi previsti. Per questo esempio, lo stack deve avere almeno tre istanze basate sul carico.

> **Note**
>
> Assicurarsi di avere tutti e tre i tipi di istanza distribuiti su più zone di disponibilità per mitigare l'impatto di eventuali interruzioni del servizio.

Best practice: Gestione delle autorizzazioni

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

È necessario disporre di credenziali di AWS per accedere alle risorse dell'account. Di seguito sono elencate alcune linee guida generali per fornire l'accesso ai dipendenti.

- In primo luogo, è consigliabile non utilizzare le credenziali root dell'account per accedere alle risorse AWS.

  Invece, crea identità IAM per i tuoi dipendenti e aggiungi le autorizzazioni che forniscono l'accesso appropriato. Ogni dipendente può quindi utilizzare le proprie credenziali per accedere alle risorse.

- I dipendenti devono avere le autorizzazioni per accedere solo alle risorse necessarie per svolgere il proprio lavoro.

  Ad esempio, gli sviluppatori di applicazioni hanno bisogno di accedere solo agli stack che eseguono le loro applicazioni.
- I dipendenti devono avere le autorizzazioni per utilizzare solo le operazioni necessarie per svolgere il proprio lavoro.

  Uno sviluppatore di applicazioni potrebbe richiedere autorizzazioni complete per uno stack di sviluppo e autorizzazioni per distribuire le app nello stack di produzione corrispondente. Probabilmente non avrà bisogno di autorizzazioni per avviare o arrestare le istanze nello stack di produzione, creare o eliminare livelli e così via.

Per ulteriori informazioni generali sulla gestione delle autorizzazioni, consulta AWS Security Credentials.

Puoi utilizzare AWS OpsWorks Stacks o IAM per gestire le autorizzazioni degli utenti. Le due opzioni non si escludono a vicenda e talvolta è opportuno utilizzare entrambe.

AWS OpsWorks Gestione delle autorizzazioni di Stacks

Ogni stack ha una pagina Permissions (Autorizzazioni) che è possibile utilizzare per concedere agli utenti l'autorizzazione di accesso allo stack e specificare le operazioni che possono eseguire. È possibile specificare le autorizzazioni di un utente impostando uno dei livelli di autorizzazioni seguenti. Ogni livello rappresenta una policy IAM che concede le autorizzazioni per un set standard di azioni.

- Deny (Nega) nega l'autorizzazione per interagire con lo stack in qualsiasi modo.
- Show (Visualizzazione) concede le autorizzazioni per visualizzare la configurazione dello stack, ma non per modificare lo stato dello stack.
- Deploy (Distribuzione) include le autorizzazioni Show (Visualizzazione) e concede all'utente anche le autorizzazioni di distribuzione delle app.
- Manage (Gestione) include le autorizzazioni Deploy (Distribuzione) e permette anche all'utente di eseguire diverse operazioni di gestione dello stack, come la creazione o l'eliminazione di istanze e livelli.

> **Note**
>
> Il livello Manage permissions non concede le autorizzazioni per un numero limitato di azioni AWS OpsWorks Stacks di alto livello, inclusa la creazione o la clonazione di stack. È necessario utilizzare una policy IAM per concedere tali autorizzazioni.

Oltre a impostare i livelli di autorizzazione, puoi anche utilizzare la pagina Autorizzazioni di uno stack per specificare se gli utenti hanno SSH/RDP and sudo/admin privilegi sulle istanze dello stack. Per ulteriori informazioni sulla gestione delle autorizzazioni di AWS OpsWorks Stacks, consulta Concessione delle autorizzazioni per stack. Per ulteriori informazioni sulla gestione dell'accesso SSH, consulta Gestione dell'accesso SSH.

Gestione delle autorizzazioni IAM

Con la gestione delle autorizzazioni IAM, utilizzi la console IAM, l'API o la CLI per allegare una policy in formato JSON a un utente che specifica esplicitamente le proprie autorizzazioni. Per ulteriori informazioni sulla gestione delle autorizzazioni IAM, consulta Che cos'è IAM? .

Raccomandazione: inizia con la gestione delle autorizzazioni di AWS OpsWorks Stacks. Se devi ottimizzare le autorizzazioni di un utente oppure concedere a un utente autorizzazioni non incluse nel livello di autorizzazioni Manage (Gestione), puoi combinare i due approcci. AWS OpsWorks Stacks valuta quindi entrambe le politiche per determinare le autorizzazioni dell'utente.

> **Important**
>
> Se un utente ha più politiche con autorizzazioni in conflitto, la negazione vince sempre. Ad esempio, supponiamo di associare una policy IAM a un utente che consente l'accesso a uno stack particolare, ma di utilizzare anche la pagina Autorizzazioni dello stack per assegnare all'utente un livello di autorizzazioni Deny. Il livello di autorizzazioni Deny (Nega) ha la precedenza e l'utente non sarà in grado di accedere allo stack. Per ulteriori informazioni, consulta la logica di valutazione delle politiche IAM.

Supponi, ad esempio, di voler fare in modo che un utente possa eseguire la maggior parte delle operazioni su uno stack, ad eccezione di aggiungere o eliminare i livelli.

- Specifica un livello di autorizzazione Manage (Gestione), che permette all'utente di eseguire la maggior parte delle operazioni di gestione dello stack, incluse la creazione e l'eliminazione di livelli.
- Allega all'utente la seguente policy gestita dal cliente, che nega le autorizzazioni all'utilizzo CreateLayere alle DeleteLayerazioni su quello stack. Puoi identificare lo stack in base al relativo Amazon Resource Name (ARN), indicato nella pagina Settings (Impostazioni) dello stack.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:CreateLayer",
        "opsworks:DeleteLayer"
      ],
      "Resource": "arn:aws:opsworks:*:*:stack/2f18b4cb-4de5-4429-a149-ff7da9f0d8ee/"
    }
  ]
}
```

Per ulteriori informazioni, incluse policy di esempio, consulta Gestione delle autorizzazioni AWS OpsWorks di Stacks allegando una policy IAM.

> **Note**
>
> Un altro modo di utilizzare la policy IAM consiste nell'impostare una condizione che limiti l'accesso allo stack ai dipendenti con un indirizzo IP o un intervallo di indirizzi specificato. Per fare in modo, ad esempio, che i dipendenti accedano agli stack solo dall'interno di un firewall aziendale, imposta una condizione che limita l'accesso all'intervallo di indirizzi IP dell'azienda. Per ulteriori informazioni, consulta la pagina relativa alle condizioni.

Best practice: gestione e distribuzione di app e libri di ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks distribuisce app e libri di cucina su ogni nuova istanza da un repository remoto. Durante il ciclo di vita di un'istanza, spesso è necessario aggiornare le app o i libri di ricette sulle istanze online dello stack per aggiungere caratteristiche, risolvere i bug e così via. Ci sono vari modi per gestire le app e i libri di ricette di uno stack, ma l'approccio utilizzato deve soddisfare i seguenti requisiti generali:

- Tutte le istanze dello stack di produzione devono avere lo stesso codice di applicazione e libro di ricette personalizzato, con eccezioni limitate come il test A/B.
- La distribuzione di un aggiornamento non deve interrompere il funzionamento del sito, anche in caso di problemi.

Questa sezione descrive le prassi raccomandate per la gestione e la distribuzione di app e libri di ricette.

Argomenti

- Mantenimento della coerenza
- Distribuzione del codice alle istanze online

Mantenimento della coerenza

In generale, è necessario mantenere uno stretto controllo sul codice dell'app o del libro di ricette in esecuzione sullo stack di produzione. Di solito, tutte le istanze devono eseguire la versione attualmente approvata del codice. Le eccezioni si verificano quando si aggiornano le app o i libri di ricette, come descritto in seguito, e in casi particolari, come l'esecuzione di test A/B.

Il codice dell'app e del libro di ricette viene distribuito da un repository di origine specificato sulle tue istanze dello stack in due modi:

- Quando avvii un'istanza, AWS OpsWorks Stacks distribuisce automaticamente l'app corrente e il codice del ricettario sull'istanza.

- Per le istanze online, è necessario distribuire manualmente il codice corrente dell'app e del libro di ricette eseguendo un comando Deploy (per le app) o un comando Update Custom Cookbooks (per i libri di ricette).

Poiché sono previsti due meccanismi di distribuzione, è fondamentale gestire il tuo codice sorgente attentamente per evitare di eseguire per errore un codice diverso su altre istanze. Ad esempio, se distribuisci app o libri di cucina da un ramo principale di Git, AWS OpsWorks Stacks distribuisce ciò che si trova in quel ramo in quel momento. Se aggiorni il codice nel ramo principale e quindi avvii una nuova istanza, quest'ultima disporrà di una versione più recente del codice rispetto alle istanze precedenti. La versione più recente potrebbe anche non essere approvata per la produzione.

Raccomandazione: Amazon S3 Archives

Per assicurarti che tutte le tue istanze abbiano la versione di codice approvata, ti consigliamo di distribuire app e libri di cucina da un archivio Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3). Ciò garantisce che il codice sia un elemento statico, un file.zip o un altro file di archivio, che deve essere aggiornato in modo esplicito. Inoltre, Amazon S3 è altamente affidabile, quindi raramente, se non mai, non sarai in grado di accedere all'archivio. Per garantire ulteriormente la coerenza, modifica in modo esplicito ogni file di archivio utilizzando una convenzione di denominazione o utilizzando il controllo delle versioni di Amazon S3, che fornisce un audit trail e un modo semplice per ripristinare una versione precedente.

Ad esempio, puoi creare una pipeline di distribuzione utilizzando uno strumento come Jenkins. Dopo aver eseguito il commit e il test del codice che desideri distribuire, crea un file di archivio e caricalo su Amazon S3. Tutte le distribuzioni delle app o gli aggiornamenti dei libri di ricette determinano l'installazione del codice in quel file di archivio e ogni istanza avrà lo stesso codice.

Raccomandazione: repository Git o Subversion

Se preferisci utilizzare un repository Git o Subversion, non distribuire dal ramo principale. Al contrario, aggiungi un tag alla versione approvata e specifica che tale versione è l'origine dell'app o del libro di ricette.

Distribuzione del codice alle istanze online

AWS OpsWorks Stacks non distribuisce automaticamente il codice aggiornato nelle istanze online. È necessario eseguire questa operazione manualmente, il che comporta le seguenti sfide:

- Distribuire l'aggiornamento in modo efficiente senza compromettere la capacità del sito di gestire le richieste dei clienti durante il processo di distribuzione.
- Gestire una distribuzione non riuscita a causa di problemi con l'app o il libro di ricette da distribuire o di anomalie associate al processo di distribuzione stesso.

L'approccio più semplice consiste nell'esecuzione di un comando Deploy predefinito (per le app) o un comando Update Custom Cookbooks (per i libri di ricette), che distribuisce l'aggiornamento in ogni istanza simultaneamente. Questo approccio è semplice e veloce, ma non vi è alcun margine di errore. Se la distribuzione ha esito negativo o il codice aggiornato genera problemi, ogni istanza nello stack di produzione potrebbe esserne interessata e pertanto il sito potrebbe potenzialmente non funzionare o venire disabilitato fino alla risoluzione del problema o al ritorno alla versione precedente.

Raccomandazione: utilizza una solida strategia di distribuzione, che permetta alle istanze che eseguono la versione del codice precedente di continuare a gestire le richieste fino alla conferma dell'avvenuta distribuzione e fino al trasferimento in totale sicurezza di tutto il traffico in entrata verso la nuova versione.

Le seguenti sezioni forniscono due esempi di strategie di distribuzione solide, seguite da una discussione su come gestire un database back-end durante la distribuzione. Per brevità, vengono descritti gli aggiornamenti delle app, ma è possibile utilizzare strategie simili per i libri di ricette.

Argomenti

- Utilizzo di una distribuzione in sequenza
- Utilizzo di stack separati
- Gestione di un database back-end

Utilizzo di una distribuzione in sequenza

Una distribuzione in sequenza aggiorna un'applicazione su istanze di server applicazioni online di uno stack in più fasi. In ciascuna fase, aggiorni un sottoinsieme di istanze online e verifichi che l'aggiornamento abbia esito positivo prima di avviare la fase successiva. In caso di anomalie, le istanze che ancora eseguono la versione dell'app precedente possono continuare a gestire il traffico in entrata finché non avrai risolto i problemi.

L'esempio seguente presuppone tu stia utilizzando la prassi raccomandata che prevede di distribuire le istanze di server applicazioni dello stack tra più zone di disponibilità.

Per eseguire una distribuzione in sequenza

1. Nella pagina di distribuzione delle app, scegliere Advanced (Avanzate), quindi scegliere una singola istanza del server applicazioni e distribuire l'app sull'istanza.

   Per cautela, è possibile rimuovere l'istanza dal sistema di bilanciamento del carico prima di distribuire l'app. In questo modo gli utenti rileveranno l'applicazione aggiornata solo dopo la verifica del suo corretto funzionamento. Se utilizzi Elastic Load Balancing, rimuovi l'istanza dal load balancer utilizzando la console Elastic Load Balancing, la CLI o un SDK.
2. Verificare che l'app aggiornata funzioni correttamente e che l'istanza presenti parametri di prestazione accettabili.

   Se hai rimosso l'istanza da un sistema di bilanciamento del carico Elastic Load Balancing, utilizza la console Elastic Load Balancing, la CLI o un SDK per ripristinarla. La versione dell'app aggiornata gestisce ora le richieste degli utenti.
3. Distribuire l'aggiornamento al resto delle istanze nella zona di disponibilità e verificare che funzionino correttamente e presentino parametri accettabili.
4. Ripetere la fase 3 per le altre zone di disponibilità dello stack, una zona alla volta. Se vuoi essere particolarmente cauto, ripeti i passaggi da 1 a 3.

> **Note**
>
> Se utilizzi un sistema di bilanciamento del carico Elastic Load Balancing, puoi utilizzare il relativo controllo dello stato per verificare che la distribuzione sia avvenuta correttamente. Tuttavia, imposta il percorso ping verso un'applicazione che controlla le dipendenze e verifica che tutto funzioni correttamente, non verso un file statico che si limita a confermare che il server applicazioni è in esecuzione.

Utilizzo di stack separati

Un altro approccio alla gestione delle applicazioni consiste nell'utilizzo di uno stack separato per ogni fase del ciclo di vita dell'applicazione. I diversi stack vengono talvolta definiti ambienti. Questa organizzazione ti consente di realizzare sviluppo e test su stack che non sono accessibili pubblicamente. Quando sei pronto a distribuire un aggiornamento, trasferisci il traffico degli utenti dallo stack che ospita la versione dell'applicazione attuale allo stack che ospita la versione aggiornata.

Argomenti

- Utilizzo di stack di sviluppo, gestione temporanea e produzione
- Utilizzo di una strategia di sviluppo blu-verde

Utilizzo di stack di sviluppo, gestione temporanea e produzione

L'approccio più comune usa gli stack seguenti.

Stack di sviluppo

Utilizza uno stack di sviluppo per attività quali l'implementazione di nuove caratteristiche o la correzione di bug. Uno stack di sviluppo è essenzialmente un prototipo dello stack di produzione, con gli stessi livelli, app, risorse e così via, che sono inclusi nel tuo stack di produzione. Poiché lo stack di sviluppo non deve in genere gestire lo stesso carico dello stack di produzione, puoi di solito utilizzare meno istanze o istanze più piccole.

Gli stack di sviluppo non sono pubblici e ne controlli l'accesso nel modo seguente:

- Limitando l'accesso alla rete tramite la configurazione delle regole in entrata del gruppo di sicurezza del server applicazioni o del sistema di bilanciamento del carico in modo da accettare solo le richieste in entrata da parte di indirizzi IP o intervalli di indirizzi specificati.

  Ad esempio, puoi limitare l'accesso HTTP, HTTPS e SSH a indirizzi del tuo intervallo di indirizzi aziendali.
- Controlla l'accesso alla funzionalità di gestione AWS OpsWorks dello stack di Stacks utilizzando la pagina Autorizzazioni dello stack.

  Ad esempio, puoi concedere un livello di autorizzazione Gestione al team di sviluppo e Mostra a tutti gli altri dipendenti.

Stack di gestione temporanea

Utilizza uno stack di gestione temporanea per verificare e finalizzare i candidati per uno stack di produzione aggiornato. +Una volta completato lo sviluppo, crea uno stack di gestione temporanea tramite clonazione dello stack di sviluppo. Quindi esegui la suite di test sullo stack di gestione temporanea e distribuisci gli aggiornamenti su tale stack per risolvere i problemi che si verificano.

Gli stack di gestione temporanea, inoltre, non sono pubblici; puoi controllare l'accesso alla rete e allo stack in modo analogo allo stack di sviluppo. Tieni presente che quando cloni uno

stack di sviluppo per creare uno stack di staging, puoi clonare le autorizzazioni concesse dalla gestione delle autorizzazioni di Stacks. AWS OpsWorks Tuttavia, la clonazione non influenza le autorizzazioni concesse dalle policy IAM degli utenti. È necessario utilizzare la console IAM, l'interfaccia a riga di comando o un SDK per modificare tali autorizzazioni. Per ulteriori informazioni, consulta Gestione delle autorizzazioni utente.

Stack di produzione

Lo stack di produzione è lo stack pubblico che supporta la tua applicazione corrente. Quando lo stack di gestione temporanea ha superato i test, lo promuovi a stack di produzione ed elimini il vecchio stack di produzione. Per un esempio su come Eseguire questa operazione, consulta Utilizzo di una strategia di sviluppo blu-verde.

> **Note**
>
> Invece di usare la console AWS OpsWorks Stacks per creare gli stack manualmente, crea un modello per ogni stack. AWS CloudFormation Questo approccio presenta i vantaggi seguenti:
>
> - Velocità e praticità: all'avvio del modello, crea AWS CloudFormation automaticamente lo stack, incluse tutte le istanze richieste.
> - Coerenza: archivia il modello per ogni stack nel tuo repository di origine per garantire che gli sviluppatori utilizzino lo stesso stack per lo stesso scopo.

Utilizzo di una strategia di sviluppo blu-verde

Una strategia di distribuzione blu-verde è un modo comune per utilizzare in modo efficiente gli stack separati per distribuire un aggiornamento dell'applicazione in produzione.

- L'ambiente blu è lo stack di produzione, che ospita l'applicazione corrente.
- L'ambiente verde è lo stack di gestione temporanea, che ospita l'applicazione aggiornata.

Quando sei pronto a distribuire l'app aggiornata in produzione, trasferisci il traffico degli utenti dallo stack blu a quello verde, che diventa il nuovo stack di produzione. Quindi ritira il vecchio stack blu.

L'esempio seguente descrive come eseguire una distribuzione blu-verde con AWS OpsWorks stack Stacks, insieme a Route 53 e un pool di sistemi di bilanciamento del carico Elastic Load Balancing. Prima di effettuare il passaggio, è necessario verificare quanto segue:

- L'aggiornamento dell'applicazione sullo stack verde ha superato il test ed è pronto per la produzione.
- Lo stack verde è identico a quello blu tranne per il fatto che include l'app aggiornata e non è pubblico.

  Entrambi gli stack dispongono delle stesse autorizzazioni, dello stesso numero e tipo di istanze in ciascun livello, della stessa configurazione basata sul tempo e basata sul carico e così via.
- Tutte le istanze 24 ore su 24, 7 giorni su 7, dello stack verde e le istanze basate sul tempo pianificate sono online.
- Disponi di un pool di sistemi di bilanciamento del carico Elastic Load Balancing che possono essere collegati dinamicamente a un layer in entrambi gli stack e possono essere preriscaldati per gestire il volume di traffico previsto.
- Hai utilizzato la funzionalità di routing ponderato di Route 53 per creare un set di record in una zona ospitata che include i sistemi di bilanciamento del carico in pool.
- Hai assegnato un peso diverso da zero al sistema di bilanciamento del carico collegato al livello del server applicazioni dello stack blu e un peso pari a zero ai sistemi di bilanciamento del carico non utilizzati. In questo modo il sistema di bilanciamento del carico dello stack blu gestisce tutto il traffico in entrata.

Per trasferire gli utenti allo stack verde

1. Collegare uno dei sistemi di bilanciamento del carico inutilizzati del pool al livello del server applicazioni dello stack verde. In alcuni casi, ad esempio quando si prevede traffico flash, oppure se non è possibile configurare un test di carico per aumentare progressivamente il traffico, inizializzare il sistema di bilanciamento del carico per gestire il traffico previsto.
2. Dopo che tutte le istanze dello stack verde hanno superato il controllo di integrità di Elastic Load Balancing, modifica i pesi nel set di record della Route 53 in modo che il load balancer dello stack verde abbia un peso diverso da zero e il load balancer dello stack blu abbia un peso corrispondente ridotto. È consigliabile iniziare facendo gestire una piccola percentuale di richieste allo stack verde, magari il 5%, lasciando gestire tutto il resto allo stack blu. Sono ora disponibili due stack di produzione: il verde che gestisce alcune delle richieste in entrata e il blu che gestisce il resto.
3. Monitorare i parametri di prestazione dello stack verde. Se sono accettabili, aumentare il peso dello stack verde in modo che possa gestire magari il 10% del traffico in entrata.

4. Ripetere la fase 3 finché lo stack verde non gestisce circa metà del traffico in entrata. A questo punto dovrebbero essere emersi tutti i potenziali problemi, pertanto se lo stack verde funziona in modo accettabile, è possibile completare il processo riducendo il peso dello stack blu a zero. Lo stack verde è ora il nuovo stack blu e gestisce tutto il traffico in entrata.
5. Scollegare il sistema di bilanciamento del carico dal livello del server applicazioni del vecchio stack blu e restituirlo al pool.
6. Anche se il vecchio stack blu non gestisce più le richieste degli utenti, consigliamo di conservarlo per un po' in caso di problemi con il nuovo stack blu. In questo caso, è possibile eseguire il roll back dell'aggiornamento invertendo la procedura per indirizzare il traffico in entrata al vecchio stack blu. Quando si è certi che il nuovo stack blu stia funzionando in modo accettabile, arrestare il vecchio stack blu.

Gestione di un database back-end

Se l'applicazione dipende da un database di backend, sarà necessario passare dalla vecchia applicazione alla nuova. AWS OpsWorks Stacks supporta le seguenti opzioni di database.

Livello Amazon RDS

Con un livello Amazon Relational Database Service (Amazon RDS), crei l'istanza DB RDS separatamente e poi la registri nel tuo stack. Puoi registrare un'istanza database RDS con un solo stack alla volta, ma è possibile trasferire un'istanza database RDS da uno stack a un altro.

AWS OpsWorks Stacks installa un file con i dati di connessione sui server delle applicazioni in un formato che può essere facilmente utilizzato dall'applicazione. AWS OpsWorks Stacks aggiunge inoltre le informazioni di connessione al database agli attributi di configurazione e distribuzione dello stack, a cui è possibile accedere tramite ricette. Puoi inoltre utilizzare JSON per fornire i dati di connessione alle applicazioni. Per ulteriori informazioni, consulta Connessione a un database.

L'aggiornamento di un'applicazione che dipende da un database pone due sfide fondamentali:

- Garantire che ogni transazione venga correttamente registrata durante la transizione evitando race condition tra le nuove e le vecchie versioni dell'applicazione.
- Eseguire la transizione in un modo che limiti l'impatto sulle prestazioni del sito e riduca al minimo o elimini tempi di inattività.

Quando utilizzi le strategie di distribuzione descritte in questo argomento, non potrai semplicemente scollegare il database dalla vecchia applicazione e ricollegarlo alla nuova. Entrambe le versioni dell'applicazione vengono eseguite in parallelo durante la transizione e devono disporre dell'accesso agli stessi dati. Di seguito vengono descritti due approcci alla gestione della transizione, entrambi con vantaggi e sfide.

Approccio 1: entrambe le applicazioni si connettono allo stesso database

Vantaggi

- Non si verificano tempi di inattività durante la transizione.

  Un'applicazione interrompe gradualmente l'accesso al database, mentre l'altra assume gradualmente il controllo.
- Non è necessario sincronizzare i dati tra i due database.

Sfide

- Entrambe le applicazioni accedono allo stesso database, quindi è necessario gestire l'accesso per evitare la perdita o il danneggiamento dei dati.
- Se è necessario eseguire la migrazione a un nuovo schema del database, la vecchia versione dell'applicazione deve essere in grado di utilizzarlo.

Se utilizzi stack separati, questo approccio è probabilmente più adatto ad Amazon RDS perché l'istanza non è legata in modo permanente a uno stack particolare ed è accessibile da applicazioni in esecuzione su stack diversi. Tuttavia, non potrai registrare un'istanza database RDS con più di uno stack alla volta, pertanto è necessario fornire i dati di connessione a entrambe le applicazioni, per esempio utilizzando JSON. Per ulteriori informazioni, consulta Utilizzo di una ricetta personalizzata.

Se utilizzi un aggiornamento progressivo, la vecchia e la nuova versione dell'applicazione sono ospitate sullo stesso stack, quindi puoi utilizzare un livello Amazon RDS o MySQL.

Approccio 2: fornire a ciascuna versione dell'applicazione il proprio database

Vantaggi

- Ogni versione dispone del proprio database, in modo che gli schemi non debbano essere compatibili.

Sfide

- Sincronizzare i dati tra i due database durante la transizione senza perdere o danneggiare dati.

- Garantire che la tua procedura di sincronizzazione non provochi notevoli tempi di inattività o peggiori in modo significativo le prestazioni del sito.

Se stai usando stack separati, ognuno ha il proprio database. Se stai usando una distribuzione in sequenza, puoi collegare due database allo stack, uno per ogni applicazione. Se la vecchia applicazione e quella aggiornata non dispongono di schemi di database compatibili, questo approccio è migliore.

Raccomandazione: in generale, consigliamo di utilizzare un livello Amazon RDS come database di backend dell'applicazione perché è più flessibile e può essere utilizzato per qualsiasi scenario di transizione. Per ulteriori informazioni su come gestire le transizioni, consulta la Amazon RDS User Guide.

Creazione di pacchetti di dipendenze di libri di ricette a livello locale

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi utilizzare Berkshelf per impacchettare le dipendenze dei tuoi libri di cucina localmente, caricare il pacchetto su Amazon S3 e modificare lo stack per utilizzare il pacchetto su Amazon S3 come fonte per libri di cucina. I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Le procedure dettagliate seguenti descrivono come preconfezionare i libri di cucina e le relative dipendenze in un file .zip per poi utilizzarlo come fonte per i libri di cucina per le istanze Linux in Stacks. AWS OpsWorks La prima procedura guidata descrive come creare un pacchetto di un libro di ricette. La seconda procedura guidata descrive invece come creare un pacchetto di più libri di ricette.

Prima di iniziare, installa il kit di sviluppo Chef (noto anche come Chef DK), ovvero una serie di strumenti sviluppati dalla community di Chef. Devi avere a disposizione questo strumento per poter utilizzare lo strumento a riga di comando `chef`.

Creazione di pacchetti di dipendenze a livello locale in Chef 12

In Chef 12 Linux, Berkshelf non è più installato per impostazione predefinita nelle istanze dello stack. Consigliamo di installare e usare Berkshelf su un computer di sviluppo locale per creare localmente il pacchetto delle dipendenze del libro di ricette. Carica il tuo pacchetto, con le dipendenze incluse, su Amazon S3. Modifica infine lo stack di Chef 12 Linux per utilizzare il pacchetto caricato come origine del libro di ricette. Tieni presente le seguenti differenze quando crei pacchetti di libri di ricette in Chef 12.

1. Nel computer locale, creare un libro di ricette eseguendo lo strumento a riga di comando `chef`.

   ```
   chef generate cookbook "server-app"
   ```

   Questo comando crea un libro di ricette, un Berksfile, un file `metadata.rb` e una directory di ricette e li posiziona in una cartella con lo stesso nome del libro di ricette. L'esempio seguente mostra la struttura degli elementi creati.

   ```
   server-app <-- the cookbook you've just created
       ### Berksfile
       ### metadata.rb
       ### recipes
   ```

2. In un editor di testo, modificare il Berksfile in modo che punti ai libri di ricette da cui dipenderà il libro di ricette `server-app`. In questo esempio, si desidera che `server-app` dipenda dal libro di ricette `java` disponibile nel supermercato Chef. In questo caso viene specificata la versione 1.50.0 o una versione anteriore più aggiornata, ma è possibile immettere qualsiasi versione pubblicata tra virgolette singole. Salvare le modifiche e chiudere il file.

   ```
   source 'https://supermarket.chef.io'
   cookbook 'java', '~> 1.50.0'
   ```

3. Modificare il file `metadata.rb` e aggiungere le dipendenze. Salvare le modifiche e chiudere il file.

   ```
   depends 'java' , '~> 1.50.0'
   ```

4. Passare alla directory `server-app` del libro di ricette creata automaticamente da Chef, quindi eseguire il comando `package` per creare un file `tar` del libro di ricette. Se si sta creato un pacchetto di più libri di ricette, eseguire questo comando nella directory principale in cui vengono

archiviati tutti i libri di ricette. Per creare un pacchetto di un singolo libro di ricette, eseguire il comando nella directory del libro di ricette desiderato. In questo esempio, il comando viene eseguito nella directory `server-app`.

```
berks package cookbooks.tar.gz
```

L'output è simile a quello riportato di seguito. Il file `tar.gz` viene creato nella directory locale.

```
Cookbook(s) packaged to /Users/username/tmp/berks/cookbooks.tar.gz
```

5. Nel AWS CLI, carica il pacchetto appena creato su Amazon S3. Annotare il nuovo URL del pacchetto del libro di ricette dopo averlo caricato in S3. Questo URL sarà necessario per le impostazioni dello stack.

```
aws s3 cp cookbooks.tar.gz s3://bucket-name/
```

L'output è simile a quello riportato di seguito.

```
upload: ./cookbooks.tar.gz to s3://bucket-name/cookbooks.tar.gz
```

6. In AWS OpsWorks Stacks, modifica lo stack per utilizzare il pacchetto che hai caricato come fonte del libro di cucina.

   a. Impostare l'opzione Use custom Chef Cookbooks (Utilizza i libri di ricette di Chef personalizzati) su Yes (Sì).

   b. Impostare Repository type (Tipo di repository) su S3 Archive (Archivio S3).

   c. In Repository URL (URL repository), incollare l'URL del pacchetto del libro di ricette caricato nella fase 5.

   Salva le modifiche apportate allo stack.

Creazione di pacchetti di dipendenze a livello locale per un libro di ricette

1. Nel computer locale, creare un libro di ricette utilizzando lo strumento a riga di comando chef:

```
chef generate cookbook "server-app"
```

Questo comando crea un libro di ricette e un Berksfile e li posiziona in una cartella con lo stesso nome del libro di ricette.

2. Passare alla directory del libro di ricette creata automaticamente da Chef, quindi creare un pacchetto di tutti gli elementi eseguendo il seguente comando:

```
berks package cookbooks.tar.gz
```

L'output sarà il seguente:

```
Cookbook(s) packaged to /Users/username/tmp/berks/cookbooks.tar.gz
```

3. Nel AWS CLI, carica il pacchetto appena creato su Amazon S3:

```
aws s3 cp cookbooks.tar.gz s3://bucket-name/
```

L'output sarà il seguente:

```
upload: ./cookbooks.tar.gz to s3://bucket-name/cookbooks.tar.gz
```

4. In AWS OpsWorks Stacks, modifica lo stack per utilizzare il pacchetto che hai caricato come fonte del libro di cucina.

Creazione di pacchetti di dipendenze a livello locale per più libri di ricette

Questo esempio crea due libri di ricette e crea un pacchetto delle relative dipendenze.

1. Nel computer locale, eseguire i seguenti comandi `chef` per generare due libri di ricette:

```
chef generate cookbook "server-app"
chef generate cookbook "server-utils"
```

In questo esempio, il libro di ricette server-app esegue le configurazioni Java. Pertanto è necessario aggiungere una dipendenza da Java.

2. Modificare `server-app/metadata.rb` aggiungendo una dipendenza al libro di ricette Java della community:

```
maintainer "The Authors"
```

```
maintainer_email "you@example.com"
license "all_rights"
description "Installs/Configures server-app"
long_description "Installs/Configures server-app"
version "0.1.0"
depends "java"
```

3. Indicare al Berkshelf gli elementi da inserire nei pacchetti modificando il file Berksfile nella directory principale del libro di ricette come segue:

```
source "https://supermarket.chef.io"
cookbook "server-app", path: "./server-app"
cookbook "server-utils", path: "./server-utils"
```

   La struttura di file ora appare nel seguente modo:

```
..
   ### Berksfile
   ### server-app
   ### server-utils
```

4. Infine, crea un pacchetto zip, caricalo su Amazon S3 e modifica lo stack AWS OpsWorks Stacks per utilizzare la nuova fonte per i libri di cucina. A tale scopo, eseguire le fasi da 2 a 4 descritte nell'argomento Creazione di pacchetti di dipendenze a livello locale per un libro di ricette.

Risorse aggiuntive

Per ulteriori informazioni sulla creazione di pacchetti di dipendenze dei libri di ricette, consulta le seguenti sezioni.

- Come impacchettare localmente le dipendenze dei Cookbook con Berkshelf sul blog di AWS DevOps
- Linux Chef 12 con Berkshelf sui forum AWS OpsWorks
- Berkshelf in Chef 12 sui forum AWS OpsWorks
- Installazione di libri di ricette personalizzati in questa guida
- Archivi di libri di ricette in questa guida

Stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Lo stack è l'entità Stacks di primo livello AWS OpsWorks . Rappresenta un set di istanze che desideri gestire collettivamente, generalmente perché accomunate da un unico scopo, ad esempio la distribuzione di applicazioni PHP. Oltre a essere utilizzato come container, uno stack gestisce le attività che si applicano all'intero gruppo di istanze, come la gestione delle applicazioni e dei libri di ricette.

Ad esempio, uno stack il cui scopo è servire applicazioni Web potrebbe essere analogo al seguente:

- Un set di istanze di server dell'applicazione, ciascuna delle quali gestisce una parte del traffico in entrata.
- Un'istanza del sistema di bilanciamento del carico, che distribuisce il traffico in entrata tra i server applicativi.
- Un'istanza di database, che serve come un archivio dati back-end per i server di applicazioni.

È pratica comune avere più stack che rappresentano diversi ambienti. Un tipico set di stack consiste di:

- Uno stack di sviluppo utilizzato dagli sviluppatori per aggiungere funzionalità, risolvere i bug ed eseguire altre attività di sviluppo e manutenzione.
- Uno stack di gestione temporanea per verificare gli aggiornamenti o le correzioni prima di esporli pubblicamente.
- Uno stack di produzione, ovvero la versione rivolta al pubblico che gestisce le richieste in entrata da parte degli utenti.

In questa sezione sono illustrate le nozioni di base per l'utilizzo degli stack.

Argomenti

- Migrazione degli stack da Amazon EC2 -Classic a un VPC
- Creare un nuovo stack
- Esecuzione di uno stack in un VPC
- Aggiornamento di uno stack
- Clonazione di uno stack
- Esegui i comandi AWS OpsWorks Stacks Stack
- Utilizzo di un JSON personalizzato
- Eliminazione di uno stack

Migrazione degli stack da Amazon EC2 -Classic a un VPC

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

In questo argomento viene descritto come migrare uno AWS OpsWorks Stacks stack dalla piattaforma di rete Amazon EC2 Classic a una rete Amazon Virtual Private Cloud (Amazon VPC).

Se hai creato il tuo AWS account prima del 04-12-2013, potresti avere il supporto per EC2 -Classic in alcune regioni. AWS Alcune EC2 risorse e caratteristiche di Amazon, ad esempio le reti avanzate e i tipi di istanza più recenti, possono essere utilizzate solo in un VPC. Alcune risorse possono essere condivise tra EC2 -Classic e un VPC, al contrario di altre. Per evitare interruzioni del servizio, consigliamo di migrare gli AWS OpsWorks Stacks stack su un VPC.

Argomenti

- Prerequisiti
- Migrazione di uno AWS OpsWorks Stacks stack su un VPC
- Consulta anche

Prerequisiti

Prima di iniziare, è necessario disporre di un VPC che soddisfi i requisiti di AWS OpsWorks Stacks configurazione. Per configurare sottoreti private nel tuo VPC per AWS OpsWorks Stacks, consulta Esecuzione di uno stack in un VPC questa guida. Puoi creare un VPC personalizzato utilizzando la console di gestione Amazon VPC. Per ulteriori informazioni, consulta le configurazioni e VPCs le sottoreti della console Amazon VPC nella Guida per l'utente di Amazon Virtual Private Cloud.

Per continuare con la migrazione, è necessario l'ID VPC e l'ID di sottorete che vuoi utilizzare.

Migrazione di uno AWS OpsWorks Stacks stack su un VPC

Innanzitutto, clona uno stack EC2 -Classic esistente utilizzando la console o l'API. AWS OpsWorks Stacks Quindi, sposta le risorse dello stack esistente nel nuovo stack. Avvia le nuove istanze nello stack clonato e distribuisci le app. Verificare che il nuovo stack funzioni. Infine, elimina le risorse EC2 -Classic dallo stack EC2 -Classic, quindi elimina il vecchio stack.

1. Clona lo stack EC2 -Classic esistente nel tuo VPC. La clonazione dello stack copia le impostazioni dello stack, i livelli, le app, gli utenti e le autorizzazioni degli utenti nel nuovo stack. Per ulteriori informazioni su come clonare uno stack, consulta Clonazione di uno stack in questa guida.

   Puoi anche clonare uno stack utilizzando l'API. AWS OpsWorks Stacks Quando cloni uno stack utilizzando AWS CLI o AWS SDKs, imposta il valore del `VpcId` parametro sull'ID del VPC in cui hai creato. Prerequisiti Per ulteriori informazioni, consulta CloneStack nella documentazione di riferimento dell'API AWS OpsWorks Stacks .
2. Crea nuove istanze nei livelli dello stack clonato. Assicuratevi di specificare l'ID della sottorete in cui avete creato. Prerequisiti Per ulteriori informazioni su come creare istanze in uno stack, consulta Aggiunta di un'istanza a un livello in questa guida.
3. Migra le tue risorse classiche, come gruppi EC2 di sicurezza, sistemi di bilanciamento del carico Elastic Load Balancing e indirizzi IP elastici sul tuo VPC, quindi associale allo stack clonato. Per ulteriori informazioni, consulta Migrare le risorse su un VPC nella EC2 Amazon User Guide.
4. Registra i volumi Amazon EBS e le istanze Amazon RDS su uno stack clonato. Per ulteriori informazioni sulla registrazione delle risorse su uno stack, consulta Registrazione di risorse con uno stack in questa guida.

I volumi Amazon EBS non sono associati a un VPC e puoi utilizzarli su più istanze EC2 sia negli stack -Classic che negli stack in un VPC. Puoi registrare istanze Amazon RDS in EC2 -Classic sia con stack -Classic che EC2 con stack in un VPC.

5. Avvia le istanze nello stack clonato, quindi sposta una piccola percentuale dei carichi di lavoro nello stack clonato. Ad esempio, sposta una piccola percentuale di traffico verso i sistemi di bilanciamento del carico Elastic Load Balancing nello stack clonato. Se utilizzi Amazon Route 53, consulta Instradamento del traffico a un sistema di bilanciamento del carico ELB nella Guida per gli sviluppatori di Amazon Route 53.

   Indirizza solo una piccola percentuale di traffico finché non sei sicuro che il nuovo stack sia funzionale e supporti le tue applicazioni. Lascia che il nuovo stack funzioni con una piccola percentuale di traffico per un periodo di prova, ad esempio una settimana. Dopo aver verificato che il nuovo stack funzioni, indirizza il traffico rimanente verso lo stack.

6. Dopo esserti assicurato che lo stack clonato funzioni, sposta il resto del traffico di produzione o dei carichi di lavoro nello stack clonato. Ora puoi fermare le istanze nello stack -Classic. EC2 Ti consigliamo di mantenere disponibile il vecchio stack per diverse settimane, in modo da poter riportare i carichi di lavoro nel vecchio stack in caso di problemi con il nuovo stack nelle settimane successive alla migrazione.

7. Quando il nuovo stack funziona da diverse settimane, elimina le istanze nello stack -Classic. EC2 Per ulteriori informazioni su come eliminare istanze, consulta Eliminazione delle AWS OpsWorks istanze di Stacks in questa guida.

   > **Important**
   >
   > Non utilizzare la EC2 console o l'API Amazon per interrompere o eliminare AWS OpsWorks le istanze.

8. Elimina le app nello stack EC2 -Classic. Per ulteriori informazioni sull'eliminazione delle app, consulta Per eliminare l'app dallo stack in questa guida.

9. Elimina lo stack EC2 -Classic. Per ulteriori informazioni su come eliminare uno stack, consulta Eliminazione di uno stack in questa guida.

Consulta anche

- Migrazione da EC2 -Classic a un VPC

- Guida al debugging e alla risoluzione dei problemi
- Esecuzione di uno stack in un VPC

Creare un nuovo stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per creare un nuovo stack, nella dashboard AWS OpsWorks Stacks, fai clic su Aggiungi stack. Puoi quindi utilizzare la pagina Add Stack (Aggiungi stack) per configurare lo stack. Al termine, fai clic su Add Stack (Aggiungi stack).

Argomenti

- Scegliere il tipo di stack da creare
- Opzioni di base
- Opzioni avanzate

Scegliere il tipo di stack da creare

Prima di creare uno stack, devi decidere quale tipo desideri creare. Per informazioni, consulta la seguente tabella.

Se vuoi creare...	Crea questo tipo di stack se vuoi...	Per informazioni su come fare, segui queste istruzioni:
Uno stack di esempio	Esplora le basi di AWS OpsWorks con uno stack Chef 12 basato su Linux e un'app Node.js di esempio.	Nozioni di base: esempio

Se vuoi creare...	Crea questo tipo di stack se vuoi...	Per informazioni su come fare, segui queste istruzioni:
Uno stack Chef 12 basato su Linux	Crea uno stack basato su Linux che utilizzi l'ultima versione di Chef supportata da AWS. OpsWorks Scegli questa opzione se sei un utente Chef esperto che desidera trarre vantaggio dall'ampia selezione di libri di ricette della community o scrivere i propri libri di ricette. Per ulteriori informazioni, consulta Chef 12 Linux.	Nozioni di base: Linux
Uno stack Chef 12.2 basato su Windows	Creare uno stack basato su Windows.	Nozioni di base: Windows
Uno stack Chef 11.10 basato su Linux	Creare questo stack se la tua organizzazione richiede l'uso di Chef 11.10 con Linux per la compatibilità con le versioni precedenti.	Nozioni di base sugli stack Linux Chef 11

Opzioni di base

La pagina Add Stack (Aggiungi stack) contiene le seguenti opzioni di base.

Stack name (Nome stack)

(Obbligatorio) Un nome utilizzato per identificare lo stack nella console Stacks. AWS OpsWorks Non è necessario che il nome sia univoco. AWS OpsWorks Stacks genera anche uno stack ID, che è un GUID che identifica in modo univoco lo stack. Ad esempio, con i comandi di AWS CLI, come update-stack, utilizzi l'ID stack per identificare lo stack specifico. Dopo aver creato uno stack, puoi trovare il relativo ID scegliendo prima Stack nel riquadro di navigazione, quindi Stack Settings (Impostazioni stack). L'ID è etichettato come ID. OpsWorks

Regione

(Obbligatorio) La regione AWS in cui le istanze saranno lanciate.

VPC

(Facoltativo) L'ID del VPC in cui deve essere lanciato lo stack. Tutte le istanze saranno lanciate in questo VPC e non potrai modificare l'ID in un secondo momento.

- Se il tuo account supporta la EC2 versione classica, puoi specificare No VPC (il valore predefinito) se non desideri utilizzare un VPC.

  Per ulteriori informazioni su EC2 Classic, consulta Piattaforme supportate.

- Se l'account non supporta EC2 Classic, è necessario specificare un VPC.

  L'impostazione predefinita è VPC predefinito, che combina la facilità d'uso di EC2 Classic con i vantaggi delle funzionalità di rete VPC. Se desideri eseguire il tuo stack in un normale VPC, devi crearlo utilizzando la console VPC, l'API o l'interfaccia a riga di comando. Per ulteriori informazioni su come creare un VPC per uno stack AWS OpsWorks Stacks, consulta Esecuzione di uno stack in un VPC. Per informazioni generali, consulta Amazon Virtual Private Cloud.

Default Availability Zone/Default subnet (Zona di disponibilità predefinita/Sottorete predefinita)

(Facoltativo) Questa impostazione varia a seconda del fatto che tu stia creando lo stack in un VPC o meno:

- Se il tuo account supporta la EC2 versione Classic e imposti VPC su No VPC, questa impostazione è denominata Zona di disponibilità predefinita, che specifica la zona di disponibilità AWS predefinita in cui verranno lanciate le istanze.
- Se il tuo account non supporta EC2 Classic o scegli di specificare un VPC, questo campo è denominato Sottorete predefinita, che specifica la sottorete predefinita in cui verranno avviate le istanze. Puoi avviare un'istanza in altre sottoreti sostituendo questo valore durante la creazione dell'istanza. Ogni sottorete è associata a una zona di disponibilità.

Puoi fare in modo che AWS OpsWorks Stacks avvii un'istanza in una zona di disponibilità o sottorete diversa sovrascrivendo questa impostazione al momento della creazione dell'istanza.

Per ulteriori informazioni su come eseguire uno stack in un VPC, consulta Esecuzione di uno stack in un VPC.

Sistema operativo predefinito

(Facoltativo) Il sistema operativo installato per impostazione predefinita in ciascuna istanza. Sono disponibili le seguenti opzioni:

- Uno dei sistemi operativi Linux integrati.
- Microsoft Windows Server 2012 R2.
- Un'AMI personalizzata, basata su uno dei sistemi operativi supportati.

  Se selezioni Use custom AMI (Usa AMI personalizzata), il sistema operativo viene determinato da un'AMI personalizzata che specifichi durante la creazione delle istanze. Per ulteriori informazioni, consulta Utilizzo di Custom AMIs.

Per ulteriori informazioni sui sistemi operativi disponibili, consulta AWS OpsWorks Sistemi operativi impilati.

> **Note**
>
> Puoi sostituire l'impostazione predefinita del sistema operativo al momento della creazione di un'istanza. Tuttavia, non puoi sostituire un sistema operativo Linux con uno Windows o un sistema operativo Windows con uno Linux.

Default SSH key (Chiave SSH predefinita)

(Facoltativo) Una coppia di EC2 chiavi Amazon della regione dello stack. Il valore predefinito è nessuno. Se specifichi una coppia di chiavi, AWS OpsWorks Stacks installa la chiave pubblica sull'istanza.

- Con le istanze Linux, puoi utilizzare la chiave privata con un client SSH per accedere alle istanze dello stack.

  Per ulteriori informazioni, consulta Accesso con SSH.
- Con le istanze Windows, puoi utilizzare la chiave privata con la EC2 console Amazon o la CLI per recuperare la password dell'amministratore di un'istanza.

  Puoi quindi utilizzare la password con un client RDP per accedere all'istanza come Amministratore. Per ulteriori informazioni, consulta Accesso con RDP.

Per ulteriori informazioni su come gestire le chiavi SSH, consulta Gestione dell'accesso SSH.

> **Note**
>
> Puoi ignorare questa impostazione specificando una coppia di chiavi diversa, o nessuna coppia di chiavi, al momento della creazione di un'istanza.

Chef version (Versione di Chef)

Questa opzione mostra la versione di Chef scelta.

Per ulteriori informazioni sulle versioni di Chef, consulta Versione di Chef.

Utilizza i libri di ricette personalizzati di Chef

Da utilizzare se desideri installare i tuoi libri di ricette Chef personalizzati sulle istanze dello stack.

Per Chef 12, l'impostazione predefinita è Yes (Sì). Per Chef 11, l'impostazione predefinita è No. L'opzione Sì mostra diverse impostazioni aggiuntive che forniscono a AWS OpsWorks Stacks le informazioni necessarie per distribuire i libri di cucina personalizzati dal loro repository alle istanze dello stack, come l'URL del repository. I dettagli dipendono dal repository utilizzato per i tuoi libri di ricette. Per ulteriori informazioni, consulta Installazione di libri di ricette personalizzati.

Stack color (Colore stack)

(Facoltativo) La tonalità utilizzata per rappresentare lo stack sulla console Stacks. AWS OpsWorks Puoi utilizzare colori diversi per stack diversi per distinguere, ad esempio, tra sviluppo, gestione temporanea e produzione di stack.

Stack tags (Tag stack)

È possibile applicare tag a livello di stack e livello. Quando crei un tag, lo stai applicando a ogni risorsa all'interno della struttura con tag. Ad esempio, se applichi un tag a uno stack, lo applichi a ogni livello e, all'interno di ogni livello, a ogni istanza, volume Amazon EBS o sistema di bilanciamento del carico Elastic Load Balancing nel layer. Per ulteriori informazioni su come attivare i tag e utilizzarli per tracciare e gestire i costi delle risorse AWS OpsWorks Stacks, consulta Utilizzo dei tag di allocazione dei costi e Attivazione dei tag di allocazione dei costi definiti dall'utente nella Guida per l'utente di Billing and Cost Management. Per ulteriori informazioni sull'assegnazione di tag in AWS OpsWorks Stack, consulta. Tag

Opzioni avanzate

Per le impostazioni avanzate, fai clic su Advanced (Avanzate) **>>** per visualizzare le sezioni Advanced options (Opzioni avanzate) e Security (Sicurezza).

Nella sezione Advanced options (Opzioni avanzate) sono disponibili le seguenti opzioni:

Default root device type (Tipo di dispositivo radice predefinito)

Determina il tipo di storage da utilizzare per il volume root dell'istanza. Per ulteriori informazioni, consulta Storage.

- Per impostazione predefinita, gli stack Linux utilizzano un volume root supportato da Amazon EBS, ma puoi anche specificare un volume root supportato dall'archivio delle istanze.
- Gli stack Windows devono utilizzare un volume root supportato da Amazon EBS.

Ruolo IAM

(Facoltativo) Il ruolo AWS Identity and Access Management (IAM) dello stack, utilizzato da AWS OpsWorks Stack per interagire con AWS per conto tuo.

Default IAM instance profile (Profilo istanza IAM predefinito)

(Facoltativo) Il ruolo IAM predefinito da associare alle EC2 istanze Amazon dello stack. Questo ruolo concede autorizzazioni alle applicazioni in esecuzione sulle istanze dello stack per accedere alle risorse AWS, come i bucket S3.

- Per concedere autorizzazioni specifiche alle applicazioni, scegli un profilo dell'istanza (ruolo) esistente con le policy appropriate.
- Inizialmente, il ruolo del profilo non concede autorizzazioni, ma puoi utilizzare la console IAM, l'API o la CLI per allegare le policy appropriate. Per ulteriori informazioni, consulta Specificazione delle autorizzazioni per le app in esecuzione su istanze EC2 .

API endpoint region (Regione endpoint API)

Questa impostazione corrisponde al valore dalla regione che scegli nelle impostazioni di base dello stack. Puoi scegliere tra i seguenti endpoint regionali.

- US East (N. Virginia) Region
- US East (Ohio) Region
- US West (Oregon) Region
- Regione Stati Uniti occidentali (California settentrionale)

- Solo regione (API) del Canada (Centrale); non disponibile per gli stack creati in AWS Management Console
- Regione Asia Pacifico (Mumbai)
- Regione Asia Pacifico (Singapore)
- Regione Asia Pacifico (Sydney)
- Regione Asia Pacifico (Tokyo)
- Regione Asia Pacifico (Seoul)
- Regione Europa (Francoforte)
- Regione Europa (Irlanda)
- Regione Europa (Londra)
- Regione Europa (Parigi)
- Regione Sud America (San Paolo)

Gli stack creati in un endpoint API non sono disponibili in altri endpoint API. Poiché gli utenti AWS OpsWorks Stacks sono anche specifici della regione, se desideri che gli utenti di AWS OpsWorks Stacks in una di queste aree di endpoint gestiscano gli stack in un'altra regione di endpoint, devi importare gli utenti nell'endpoint a cui sono associati gli stack. Per ulteriori informazioni sull'importazione di utenti, consulta Importazione di utenti in stacks. AWS OpsWorks

Hostname theme (Tema del nome host)

(Facoltativo) Una stringa utilizzata per generare un nome host predefinito per ogni istanza. Il valore predefinito è Layer Dependent (Dipendente dal livello), che usa il nome breve del livello dell'istanza e aggiunge un numero univoco a ogni istanza. Ad esempio, la radice del tema Load Balancer (Sistema di bilanciamento del carico) dipendente dal ruolo è "lb". La prima istanza che aggiungi al livello viene denominata "lb1", la seconda "lb2" e così via.

OpsWorks Versione dell'agente

(Facoltativo) Se aggiornare automaticamente l'agente AWS OpsWorks Stacks quando è disponibile una nuova versione o utilizzare una versione dell'agente specificata e aggiornarla manualmente. Questa funzionalità è disponibile sugli stack Chef 11.10 e Chef 12. L'impostazione predefinita è Manual update (Aggiornamento manuale), configurata sulla versione più recente dell'agente.

AWS OpsWorks Stacks installa un agente su ogni istanza che comunica con il servizio e gestisce attività come l'avvio delle esecuzioni di Chef in risposta agli eventi del ciclo di vita. Questo agente

viene aggiornato regolarmente. Sono disponibili due opzioni per specificare la versione dell'agente per il tuo stack.

- Aggiornamento automatico: AWS OpsWorks Stack installa automaticamente ogni nuova versione dell'agente nelle istanze dello stack non appena l'aggiornamento è disponibile.
- Aggiornamento manuale: AWS OpsWorks Stack installa la versione dell'agente specificata sulle istanze dello stack.

  AWS OpsWorks Stacks pubblica un messaggio nella pagina dello stack quando è disponibile una nuova versione dell'agente, ma non aggiorna le istanze dello stack. Per aggiornare l'agente, devi aggiornare manualmente le impostazioni dello stack per specificare una nuova versione dell'agente e AWS OpsWorks Stacks aggiornerà quindi le istanze dello stack.

È possibile sovrascrivere l'impostazione predefinita della versione OpsWorks dell'agente per una particolare istanza aggiornandone la configurazione. In questo caso, l'impostazione dell'istanza ha la precedenza. Ad esempio, supponiamo che l'impostazione predefinita sia Auto-update (Aggiornamento automatico), ma specifichi Manual update (Aggiornamento manuale) per una determinata istanza. Quando AWS OpsWorks Stacks rilascia una nuova versione dell'agente, aggiornerà automaticamente tutte le istanze dello stack tranne quella impostata su Aggiornamento manuale. Per installare una nuova versione dell'agente in tale istanza, devi aggiornare manualmente la relativa configurazione e specificare una nuova versione.

> **Note**
>
> La console visualizza i numeri di versione dell'agente abbreviati. Per visualizzare i numeri di versione completi, chiama il describe-agent-versionscomando AWS CLI o i metodi API o SDK equivalenti. Questi restituiscono i numeri di versione completi per le versioni disponibili dell'agente.

JSON personalizzato

(Facoltativo) Uno o più attributi personalizzati, con il formato di una struttura JSON. Questi attributi sono uniti agli attributi di configurazione e distribuzione dello stack installati in ogni istanza e possono essere utilizzati dalle ricette. Puoi utilizzare un JSON personalizzato, ad esempio per personalizzare le impostazioni di configurazione sostituendo gli attributi integrati che specificano le impostazioni predefinite. Per ulteriori informazioni, consulta Utilizzo di un JSON personalizzato.

La sicurezza ha un'opzione, Usa i gruppi OpsWorks di sicurezza, che ti consente di specificare se associare i gruppi di sicurezza integrati di AWS OpsWorks Stack ai livelli dello stack.

AWS OpsWorks Stack fornisce un set standard di gruppi di sicurezza integrati, uno per ogni livello, ognuno associato per impostazione predefinita al proprio livello. L'utilizzo dei gruppi di OpsWorks sicurezza consente invece di fornire gruppi di sicurezza personalizzati. Per ulteriori informazioni, consulta Utilizzo dei gruppi di sicurezza.

Usa i gruppi OpsWorks di sicurezza ha le seguenti impostazioni:

- Sì - AWS OpsWorks Stacks associa automaticamente il gruppo di sicurezza integrato appropriato a ogni livello (impostazione predefinita).

  Puoi associare ulteriori gruppi di sicurezza a un livello dopo che lo avrai creato, ma non puoi eliminare il gruppo di sicurezza integrato.
- No, AWS OpsWorks Stacks non associa i gruppi di sicurezza integrati ai livelli.

  È necessario creare gruppi EC2 di sicurezza appropriati e associare un gruppo di sicurezza a ogni livello creato. Tuttavia, puoi ancora associare manualmente un gruppo di sicurezza integrato a un livello al momento della creazione. I gruppi di sicurezza personalizzati sono necessari solo per quei livelli che necessitano di impostazioni personalizzate.

Tieni presente quanto segue:

- Se Usa gruppi OpsWorks di sicurezza è impostato su Sì, non è possibile limitare le impostazioni di accesso alle porte di un gruppo di sicurezza predefinito aggiungendo un gruppo di sicurezza più restrittivo a un livello. Con più gruppi di sicurezza, Amazon EC2 utilizza le impostazioni più permissive. Inoltre, non puoi creare impostazioni più restrittive modificando la configurazione del gruppo di sicurezza integrato. Quando crei uno stack, AWS OpsWorks Stacks sovrascrive le configurazioni dei gruppi di sicurezza integrati con le impostazioni standard, quindi tutte le modifiche apportate andranno perse alla successiva creazione di uno stack. Se un livello richiede impostazioni di gruppo di sicurezza più restrittive rispetto al gruppo di sicurezza integrato, imposta Usa gruppi di OpsWorks sicurezza su No, crea gruppi di sicurezza personalizzati con le tue impostazioni preferite e assegnali ai livelli al momento della creazione.
- Se elimini accidentalmente un gruppo di sicurezza AWS OpsWorks Stacks e desideri ricrearlo, è necessario che sia un duplicato esatto dell'originale, comprese le lettere maiuscole del nome del gruppo. Anziché ricreare il gruppo manualmente, ti consigliamo di far eseguire l'attività automaticamente da AWS OpsWorks Stacks. Basta creare un nuovo stack nella stessa regione

AWS e il VPC, se AWS OpsWorks presente, e Stacks ricreerà automaticamente tutti i gruppi di sicurezza integrati, incluso quello che hai eliminato. È quindi possibile eliminare lo stack se non è più necessario. I gruppi di sicurezza verranno conservati.

Esecuzione di uno stack in un VPC

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi controllare l'accesso utente a istanze di uno stack creandolo in un virtual private cloud (VPC). Ad esempio, supponi di non volere che gli utenti abbiano accesso diretto ai server applicazioni o ai database dello stack e richiedere invece che tutto il traffico pubblico venga incanalato attraverso un sistema di bilanciamento del carico elastico.

Procedura di base per l'esecuzione di uno stack in un VPC:

1. Crea un VPC configurato in modo appropriato utilizzando la console o l'API Amazon VPC o un modello. AWS CloudFormation
2. Indicazione dell'ID VPC al momento della creazione dello stack.
3. Avvio delle istanze dello stack nella sottorete appropriata.

Di seguito viene descritto brevemente come VPCs funziona in Stacks. AWS OpsWorks

> **Important**
>
> Se utilizzi la caratteristica Endpoint VPC, tieni presente che ogni istanza dello stack deve essere in grado di completare le seguenti azioni in Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3):
>
> - Installazione dell'agente dell'istanza.
> - Installazione degli asset, come Ruby.

- Caricamento dei log di esecuzione di Chef.
- Recupero dei comandi dello stack.

Per permettere queste operazioni, devi assicurarti che le istanze dello stack abbiano accesso ai bucket seguenti, corrispondenti alla regione dello stack. In caso contrario, le operazioni precedenti non riusciranno.
Per Chef 12 Linux e Chef 12.2 Windows, i bucket sono i seguenti.

Bucket di agenti	Bucket di asset	Bucket di log	Bucket DNA
• opsworks-instance-agent-sa-east-1 • opsworks-instance-agent-ap-south-1 • opsworks-instance-agent-ap-northeast-1 • opsworks-instance-agent-ap-northeast-2 • opsworks-instance-agent-ap-southeast-1 • opsworks-instance-agent-ap-southeast-2 • opsworks-instance-agent-ca-central-1 • opsworks-instance-agent-eu-central-1 • opsworks-instance-agent-eu-west-1 • opsworks-instance-agent-eu-ovest-2	• opsworks-instance-assets-us-est-2 • opsworks-instance-assets-us-east-1 • opsworks-instance-assets-ap-south-1 • opsworks-instance-assets-ap-northeast-1 • opsworks-instance-assets-ap-northeast-2 • opsworks-instance-assets-ap-southeast-1 • opsworks-instance-assets-ap-southeast-2 • opsworks-instance-assets-ca-central-1 • opsworks-instance-assets-eu-central-1 • opsworks-instance-assets-eu-west-1	• opsworks-us-east-2 registri • opsworks-us-east-1 registro • opsworks-ap-south-1 registro • opsworks-ap-northeast-1 registro • opsworks-ap-northeast-2 registri • opsworks-ap-southeast-1 registro • opsworks-ap-southeast-2 registri • opsworks-ca-central-1 registro • opsworks-eu-central-1 registro • opsworks-eu-west-1 registro • opsworks-eu-west-2 registri • opsworks-eu-west-3 registri • opsworks-sa-east-1 registro	• opsworks-us-east-2-DNA • opsworks-us-east-1 DNA • opsworks-ap-south-1-dna • opsworks-ap-northeast-1-dna • opsworks-ap-northeast-2-dna • opsworks-ap-southeast-1 DNA • opsworks-ap-southeast-2-dna • opsworks-ca-central-1 DNA • opsworks-eu-central-1-dna • opsworks-eu-west-1-dna • opsworks-eu-west-2-dna • opsworks-eu-west-3-dna • opsworks-sa-east-1 DNA • opsworks-us-west-1-dna • opsworks-us-west-2-dna

Bucket di agenti	Bucket di asset	Bucket di log	Bucket DNA
• opsworks-instance-agent-eu-west-3 • opsworks-instance-agent-us-east-1 • opsworks-instance-agent-us-est-2 • opsworks-instance-agent-us-west-1 • opsworks-instance-agent-us-ovest-2	• opsworks-instance-assets-eu-ovest-2 • opsworks-instance-assets-eu-west-3 • opsworks-instance-assets-sa-east-1 • opsworks-instance-assets-us-west-1 • opsworks-instance-assets-us-ovest-2	• opsworks-us-west-1 registro • opsworks-us-west-2 registri	

Per Chef 11.10 e versioni precedenti per Linux, i bucket sono i seguenti. Gli stack Chef 11.4 non sono supportati negli endpoint regionali al di fuori della regione Stati Uniti orientali (Virginia settentrionale).

Bucket di agenti	Bucket di asset	Bucket di log	Bucket DNA
• opsworks-instance-agent-us-est-2 • opsworks-instance-agent-us-east-1 • opsworks-instance-agent-ap-south-1 • opsworks-instance-agent-ap-northeast-1 • opsworks-instance-agent-ap-northeast-2 • opsworks-instance-agent-ap-southeast-1 • opsworks-instance-agent-ap-southeast-2 • opsworks-instance-agent-ca-central-1 • opsworks-instance-agent-eu-central-1 • opsworks-instance-agent-eu-west-1	• opsworks-instance-assets-us-est-2 • opsworks-instance-assets-us-east-1 • opsworks-instance-assets-ap-south-1 • opsworks-instance-assets-ap-northeast-1 • opsworks-instance-assets-ap-northeast-2 • opsworks-instance-assets-ap-southeast-1 • opsworks-instance-assets-ap-southeast-2 • opsworks-instance-assets-ca-central-1 • opsworks-instance-assets-eu-central-1 • opsworks-instance-assets-eu-west-1	• prod_stage-log	• prod_stage-dna

Bucket di agenti	Bucket di asset	Bucket di log	Bucket DNA
• opsworks-instance-agent-eu-ovest-2 • opsworks-instance-agent-eu-west-3 • opsworks-instance-agent-us-east-1 • opsworks-instance-agent-us-west-1 • opsworks-instance-agent-us-ovest-2	• opsworks-instance-assets-eu-ovest-2 • opsworks-instance-assets-eu-west-3 • opsworks-instance-assets-sa-east-1 • opsworks-instance-assets-us-west-1 • opsworks-instance-assets-us-ovest-2		

Per ulteriori informazioni, consulta Endpoint VPC.

> **Note**
>
> Affinché AWS OpsWorks gli Stacks si connettano agli endpoint VPC abilitati, devi anche configurare il routing per il tuo NAT o IP pubblico, poiché AWS OpsWorks l'agente Stacks richiede ancora l'accesso all'endpoint pubblico.

Argomenti

- Nozioni di base sui VPC
- Crea un VPC per uno stack di pile AWS OpsWorks

Nozioni di base sui VPC

Per una discussione dettagliata di VPCs, consulta Amazon Virtual Private Cloud. In breve, un VPC è costituito da una o più sottoreti, ognuna delle quali contiene una o più istanze. A ogni sottorete è associata una tabella di routing che indirizza il traffico in uscita in base all'indirizzo IP di destinazione.

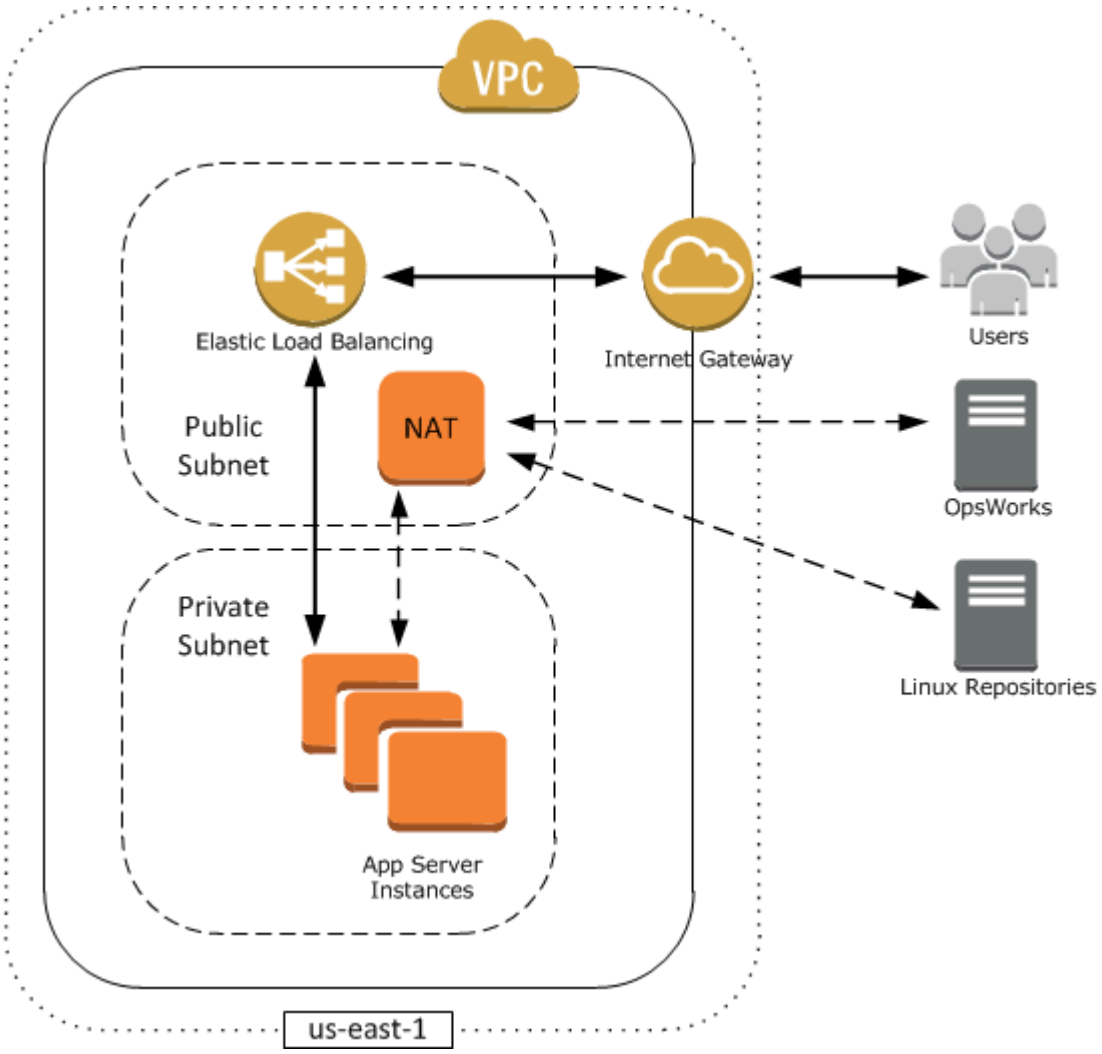
- Le istanze all'interno di un VPC possono comunicare tra loro per impostazione predefinita, indipendentemente dalla rispettiva sottorete. Tuttavia, le modifiche alle liste di controllo degli accessi alla rete (ACLs), alle politiche dei gruppi di sicurezza o l'utilizzo di indirizzi IP statici possono interrompere questa comunicazione.
- Le sottoreti le cui istanze sono in grado di comunicare con Internet vengono definite sottoreti pubbliche.
- Le sottoreti le cui istanze possono comunicare solo con altre istanze nel VPC e non direttamente con Internet vengono definite sottoreti private.

AWS OpsWorks Stacks richiede che il VPC sia configurato in modo che ogni istanza dello stack, incluse le istanze nelle sottoreti private, abbia accesso ai seguenti endpoint:

- Uno degli endpoint del servizio AWS OpsWorks Stacks elencati nella sezione «Region Support» di. Guida introduttiva a AWS OpsWorks Stacks
- Uno dei seguenti endpoint del servizio di istanza, utilizzato dall'agente Stacks. AWS OpsWorks L'agente viene eseguito sulle le istanze del cliente gestite per lo scambio dei dati con il servizio.
  - opsworks-instance-service.us-east-2. amazonaws.com
  - opsworks-instance-service.us-east-1.amazonaws.com
  - opsworks-instance-service.us-west-1. amazonaws.com
  - opsworks-instance-service.us-west-2.amazonaws.com
  - opsworks-instance-service.ap-south-1.amazonaws.com
  - opsworks-instance-service. ap-northeast-1.amazonaws.com
  - opsworks-instance-service. ap-northeast-2.amazonaws.com
  - opsworks-instance-service. ap-southeast-1.amazonaws.com
  - opsworks-instance-service.ap-southeast-2.amazonaws.com
  - opsworks-instance-service.ca-central-1.amazonaws.com
  - opsworks-instance-service.eu-central-1.amazonaws.com
  - opsworks-instance-service.eu-west-1.amazonaws.com

  - opsworks-instance-service.eu-west-2.amazonaws.com
  - opsworks-instance-service.eu-west-3.amazonaws.com
- Amazon S3
- Qualsiasi repository di pacchetti da cui dipende il sistema operativo, ad esempio i repository Amazon Linux o Ubuntu Linux.
- I repository della tua app e del libro di ricette personalizzato.

Esistono molti modi diversi per configurare un VPC in modo da fornire questa connettività. Di seguito è riportato un semplice esempio di come configurare un VPC per uno stack di server di app AWS OpsWorks Stacks.



Questo VPC include diversi componenti:

Sottoreti

Il VPC ha due sottoreti, una pubblica e una privata.

- La sottorete pubblica contiene un sistema di bilanciamento del carico e un dispositivo NAT (Network Address Translation), che possono comunicare con indirizzi esterni e con le istanze nella sottorete privata.
- La sottorete privata contiene i server applicazioni, che possono comunicare con il dispositivo NAT e il sistema di bilanciamento del carico nella sottorete pubblica, ma non possono comunicare direttamente con indirizzi esterni.

Internet Gateway

Il gateway Internet permette alle istanze con indirizzi IP pubblici, come il sistema di bilanciamento del carico, di comunicare con indirizzi esterni al VPC.

Sistema di bilanciamento del carico

Il sistema di bilanciamento del carico Elastic Load Balancing riceve il traffico in ingresso dagli utenti, lo distribuisce ai server applicazioni nella sottorete privata e restituisce le risposte agli utenti.

NAT

Il dispositivo (NAT) fornisce ai server applicazioni accesso Internet limitato, utilizzato in genere per scopi come il download degli aggiornamenti software da un repository esterno. Tutte le istanze AWS OpsWorks Stacks devono essere in grado di comunicare con AWS OpsWorks Stacks e con i repository Linux appropriati. Un metodo per gestire questo problema consiste nell'inserire un dispositivo NAT con un indirizzo IP elastico (EIP) associato in una sottorete pubblica. Puoi quindi instradare il traffico in uscita dalle istanze nella sottorete privata attraverso il dispositivo NAT.

> **Note**
>
> Una sola istanza NAT crea un singolo punto di errore nel traffico in uscita della sottorete privata. Puoi migliorare l'affidabilità configurando il VPC con una coppia di istanze NAT che possano subentrare una all'altra quando una delle due è danneggiata. Per ulteriori informazioni, consulta la pagina relativa all'alta disponibilità per istanze NAT VPC. Puoi anche utilizzare un gateway NAT. Per ulteriori informazioni, consulta NAT nella Amazon VPC User Guide.

La configurazione VPC ottimale dipende dallo stack di AWS OpsWorks stack utilizzato. Di seguito vengono forniti alcuni esempi dei casi in cui puoi usare determinate configurazioni VPC. Per esempi di altri scenari VPC, consulta la pagina relativa agli scenari per l'uso di Amazon VPC.

Utilizzo di un'istanza in una sottorete pubblica

Se disponi di uno stack a istanza singola senza risorse private associate, ad esempio un'istanza Amazon RDS che non dovrebbe essere accessibile pubblicamente, puoi creare un VPC con una sottorete pubblica e inserire l'istanza in quella sottorete. Se non utilizzi un VPC predefinito, il livello dell'istanza deve assegnare un indirizzo IP elastico (EIP) all'istanza. Per ulteriori informazioni, consulta OpsWorks Nozioni di base sui livelli.

Utilizzo di risorse private

Se hai risorse che non devono essere accessibili pubblicamente, puoi creare un VPC con una sottorete pubblica e una sottorete privata. Ad esempio, in un ambiente di scalabilità automatica con bilanciamento del carico, puoi inserire tutte le EC2 istanze Amazon nella sottorete privata e il sistema di bilanciamento del carico in una sottorete pubblica. In questo modo non è possibile accedere direttamente EC2 alle istanze Amazon da Internet; tutto il traffico in entrata deve essere instradato attraverso il sistema di bilanciamento del carico.

La sottorete privata isola le istanze dall'accesso EC2 diretto degli utenti di Amazon, ma devono comunque inviare richieste in uscita ad AWS e agli archivi di pacchetti Linux appropriati. Per permettere queste richieste, puoi, ad esempio, utilizzare un dispositivo NAT (Network Address Translation) con il relativo indirizzo IP elastico (EIP) e quindi instradare il traffico in uscita delle istanze attraverso il dispositivo NAT. Puoi inserire il dispositivo NAT nella stessa sottorete pubblica del sistema di bilanciamento del carico, come mostrato nell'esempio precedente.

- Se utilizzi un database di back-end come un'istanza Amazon RDS, puoi inserire tali istanze nella sottorete privata. Per le istanze Amazon RDS, devi specificare almeno due sottoreti diverse in diverse zone di disponibilità.
- Se hai bisogno dell'accesso diretto alle istanze in una sottorete privata, ad esempio, desideri utilizzare SSH per accedere a un'istanza, puoi inserire un bastion host nella sottorete pubblica che invia tramite proxy le richieste da Internet.

Estensione della rete in AWS

Se vuoi estendere la rete nel cloud e anche accedere direttamente a Internet dal VPC, puoi creare un gateway VPN. Per ulteriori informazioni, consulta lo scenario 3: VPC con sottoreti pubbliche e private e accesso VPN hardware.

Crea un VPC per uno stack di pile AWS OpsWorks

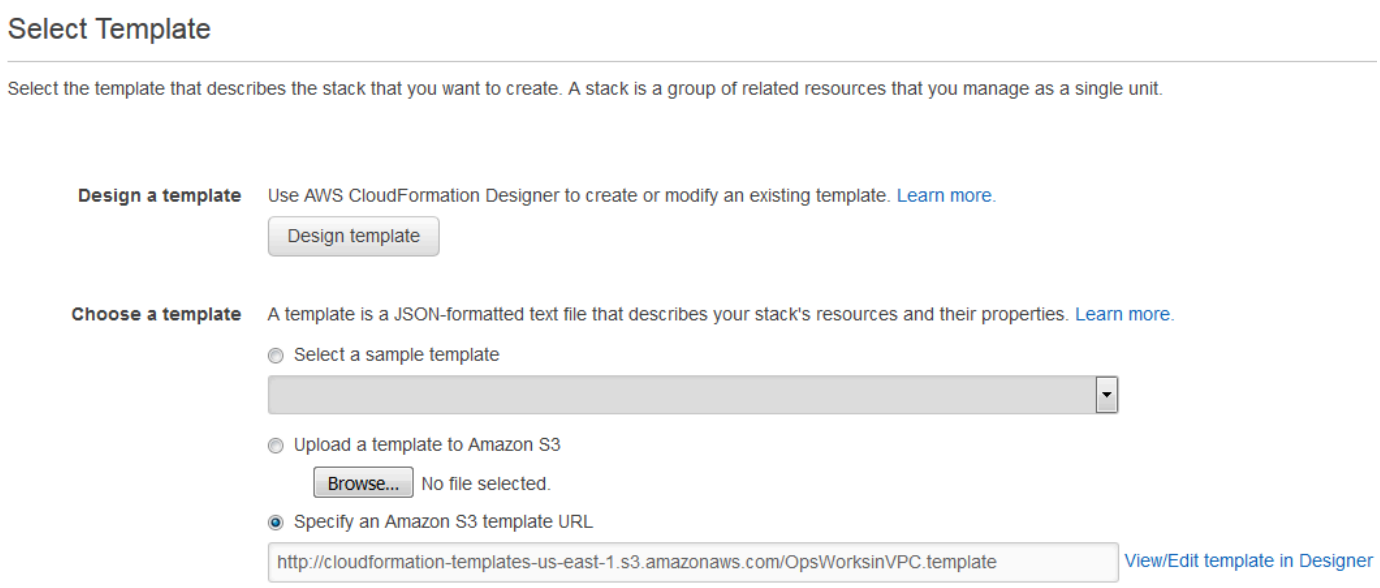
Questa sezione mostra come creare un VPC per uno stack AWS OpsWorks Stacks utilizzando un modello AWS di esempio. CloudFormation Puoi scaricare il modello nel file.zip. OpsWorks VPCtemplates Per ulteriori informazioni su come creare manualmente un VPC come quello descritto in questo argomento, consulta lo scenario 2: VPC con sottoreti pubbliche e private. Per informazioni su come configurare tabelle di routing, gruppi di sicurezza e così via, consulta il modello di esempio.

> **Note**
>
> Per impostazione predefinita, AWS OpsWorks Stacks visualizza i nomi delle sottoreti concatenando l'intervallo CIDR e la zona di disponibilità, ad esempio. `10.0.0.1/24 - us-east-1b` Per rendere i nomi più leggibili, crea un tag per ogni sottorete con Key impostato su e Value impostato sul nome della sottorete. **`Name`** AWS OpsWorks Stacks aggiunge quindi il nome della sottorete al nome predefinito. Ad esempio, la sottorete privata dell'esempio seguente ha un tag con Nome impostato su**`Private`**, che viene visualizzato come. OpsWorks `10.0.0.1/24 us-east - 1b - Private`

Puoi avviare un modello VPC utilizzando la AWS CloudFormation console con pochi passaggi. La procedura seguente utilizza il modello di esempio per creare un VPC nella regione US East (N. Virginia). Per indicazioni su come utilizzare il modello per creare un VPC in altre regioni, consulta la nota che segue la procedura.

Per creare il VPC

1. Aprire la console AWS CloudFormation, selezionare la regione US East (N. Virginia) (Stati Uniti orientali (Virginia settentrionale)) e scegliere Create Stack (Crea stack).
2. Nella pagina Select Template (Seleziona modello), selezionare Upload a template to Amazon S3 (Carica un modello in Amazon S3). Cerca il **`OpsWorksinVPC.template`** file scaricato nel OpsWorksVPCtemplatesfile.zip .Scegli Continua.



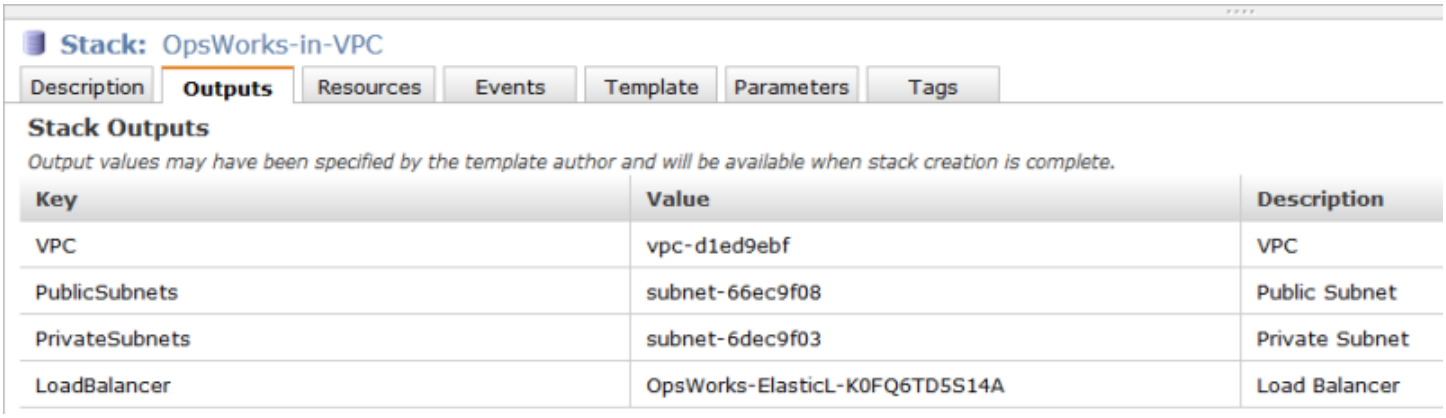
Puoi anche avviare questo stack aprendo AWS CloudFormation Sample Templates, individuando il modello VPC AWS OpsWorks Stacks e scegliendo Launch Stack.

3. Nella pagina Specify Parameters (Specifica parametri) accettare i valori predefiniti e scegliere Continue (Continua).
4. Nella pagina Add Tags (Aggiungi tag) creare un tag con Key (Chiave) impostato su **`Name`** e Value (Valore) impostato sul nome del VPC. Questo tag faciliterà l'identificazione del tuo VPC quando crei uno AWS OpsWorks stack Stacks.
5. Scegliere Continue (Continua), quindi Close (Chiudi) per avviare lo stack.

Nota: puoi creare il VPC in altre regioni utilizzando uno dei metodi seguenti.

- Vai a Utilizzo di modelli in diverse regioni, scegli la regione appropriata, individua il modello VPC AWS OpsWorks Stacks, quindi scegli Launch Stack.
- Copia il file del modello nel sistema, seleziona la regione appropriata nella console AWS CloudFormation e utilizza l'opzione Upload a template to Amazon S3 (Carica un modello in Amazon S3) della procedura guidata Create Stack (Crea stack) per caricare il modello dal sistema.

Il modello di esempio include output che forniscono il VPC, la sottorete e il load IDs balancer necessari per creare lo stack Stacks. AWS OpsWorks Puoi vederli scegliendo la scheda Output nella parte inferiore della finestra della console. AWS CloudFormation


Stack: OpsWorks-in-VPC

Description | Outputs | Resources | Events | Template | Parameters | Tags

Stack Outputs

Output values may have been specified by the template author and will be available when stack creation is complete.

Key	Value	Description
VPC	vpc-d1ed9ebf	VPC
PublicSubnets	subnet-66ec9f08	Public Subnet
PrivateSubnets	subnet-6dec9f03	Private Subnet
LoadBalancer	OpsWorks-ElasticL-K0FQ6TD5S14A	Load Balancer

Aggiornamento di uno stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver creato uno stack, puoi aggiornare la configurazione in qualsiasi momento. Nella pagina Stack, fai clic su Stack Settings (Impostazioni stack), quindi su Edit (Modifica). Viene visualizzata la pagina Settings (Impostazioni). Apporta le modifiche desiderate e fai clic su Save (Salva).

Le impostazioni sono identiche a quelle discusse nell'argomento Creare un nuovo stack. Per ulteriori dettagli, consulta tale argomento. Si noti, tuttavia, quanto segue:

- Non puoi modificare l'ID della regione o del VPC.
- Se lo stack è eseguito in un VPC, le impostazioni includono un'impostazione Default subnet (Sottorete predefinita), in cui sono elencate le sottoreti del VPC. Se lo stack non è eseguito in un VPC, l'impostazione è denominata Default Availability Zones (Zone di disponibilità predefinite), in cui sono elencate le zone di disponibilità di una regione.
- Puoi modificare il sistema operativo predefinito, ma non puoi specificare un sistema operativo Linux per uno stack Windows oppure Windows per uno stack Linux.

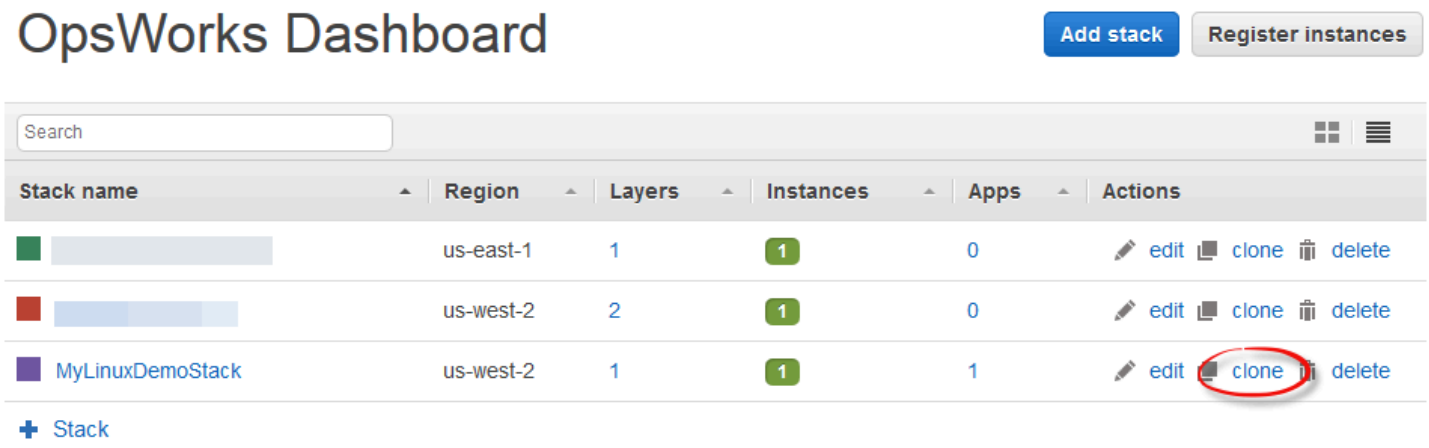
- Se modifichi una qualsiasi delle impostazioni predefinite dell'istanza, ad esempio Hostname theme (Tema del nome host) o Default SSH key (Chiave SSH predefinita), i nuovi valori si applicano solo alle nuove istanze create e non alle istanze esistenti.
- La modifica del nome modifica il nome visualizzato dalla console; non cambia il nome breve sottostante utilizzato da AWS OpsWorks Stacks per identificare lo stack.
- Prima di modificare Usa gruppi OpsWorks di sicurezza da Sì a No, ogni livello deve avere almeno un gruppo di sicurezza oltre al gruppo di sicurezza integrato del livello. Per ulteriori informazioni, consulta Modifica della configurazione di un livello OpsWorks .

  AWS OpsWorks Stacks elimina quindi i gruppi di sicurezza integrati da ogni livello.
- Se modifichi Usa i gruppi OpsWorks di sicurezza da No a Sì, AWS OpsWorks Stacks aggiunge il gruppo di sicurezza integrato appropriato a ciascun livello ma non elimina i gruppi di sicurezza esistenti.

Clonazione di uno stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Talvolta è utile creare più copie di uno stack. Ad esempio, potresti voler aggiungere ridondanza come misura di prevenzione o di disaster recovery oppure potresti utilizzare uno stack esistente come punto di partenza per un nuovo stack. L'approccio più semplice è clonare lo stack di origine. Nella dashboard AWS OpsWorks Stacks, nella colonna Azioni della riga relativa allo stack che desideri clonare, scegli clone, per aprire la pagina Clone stack.



Inizialmente, le impostazioni per lo stack clonato sono identiche a quelle dello stack di origine, tranne per il fatto che viene aggiunta la parola "copia" al nome dello stack. Per informazioni su queste impostazioni, consulta Creare un nuovo stack. Sono previste anche altre due impostazioni facoltative:

Autorizzazioni

Se l'impostazione predefinita all permissions (tutte le autorizzazioni) è selezionata, le autorizzazioni dello stack di origine vengono applicate allo stack clonato.

Web

Elenca le app distribuite sullo stack di origine. Le app dell'elenco con la casella di controllo corrispondente selezionata (impostazione predefinita) saranno distribuite sullo stack clonato.

> **Note**
>
> Non è possibile clonare uno stack da un endpoint regionale a un altro; ad esempio, non è possibile clonare uno stack dalla regione degli Stati Uniti occidentali (Oregon) (us-west-2) alla regione Asia Pacifico (Mumbai) (ap-south-1).

Dopo aver finalizzato le impostazioni, scegli Clone stack. AWS OpsWorks Stacks crea un nuovo stack composto dai livelli dello stack di origine e, facoltativamente, dalle app e dalle autorizzazioni. I livelli hanno la stessa configurazione degli originali, soggetti a eventuali modifiche apportate. Tuttavia, con la clonazione non vengono create istanze. È necessario aggiungere un set di istanze appropriato per ogni livello dello stack clonato e quindi avviarle. Come per qualsiasi stack, su uno stack clonato puoi eseguire le normali attività di gestione, ad esempio aggiunta, eliminazione o modifica di livelli o aggiunta e distribuzione di app.

Per rendere operativo lo stack clonato, avvia le istanze. AWS OpsWorks Stacks imposta e configura ogni istanza in base all'appartenenza al relativo livello. Inoltre, distribuisce qualsiasi applicazione, proprio come con un nuovo stack.

Esegui i comandi AWS OpsWorks Stacks Stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks fornisce una serie di comandi stack, che puoi utilizzare per eseguire una serie di operazioni sulle istanze di uno stack. Per eseguire un comando dello stack, fai clic su Run Command (Esegui comando) nella pagina Stack. Quindi, scegli il comando appropriato, specifica eventuali opzioni e premi il pulsante in basso a destra, che verrà etichettato con il nome del comando.

> **Note**
>
> AWS OpsWorks Stacks supporta anche un set di comandi di distribuzione, che puoi utilizzare per gestire la distribuzione delle app. Per ulteriori informazioni, consulta Distribuzione di app.

Puoi eseguire i seguenti comandi su qualsiasi stack.

Update Custom Cookbooks (Aggiorna libri di ricette personalizzati)

Aggiorna i libri di ricette personalizzati delle istanze con la versione corrente del repository. Questo comando non consente di eseguire alcuna ricetta. Per eseguire le ricette aggiornate, puoi utilizzare un comando dello stack `Execute Recipes`, `Setup` o `Configure` oppure puoi ridistribuire la tua applicazione per eseguire le ricette di ditribuzione. Per ulteriori informazioni sui libri di ricette personalizzati, consulta Libri di ricette e ricette.

Execute Recipes (Esegui ricette)

Esegue un determinato set di ricette sulle istanze. Per ulteriori informazioni, consulta Esecuzione manuale delle ricette.

Configurazione

Esegue le ricette di impostazione delle istanze.

Configura

Esegue le ricette di configurazione delle istanze.

> **Note**
>
> Per usare Setup (Installa) o Configure (Configura) allo scopo di eseguire ricette su un'istanza, le ricette devono essere assegnate all'evento del ciclo di vita corrispondente per il livello dell'istanza. Per ulteriori informazioni, consulta Esecuzione di ricette.

Puoi eseguire i seguenti comandi dello stack solo su stack basati su Linux.

Install Dependencies (Installa dipendenze)

Installa i pacchetti delle istanze. A partire da Chef 12 questo comando non è più disponibile.

Update Dependencies (Aggiorna dipendenze)

(Solo Linux. A partire da Chef 12, questo comando non è disponibile.) Installa gli aggiornamenti standard del sistema operativo e del pacchetto. I dettagli dipendono dal sistema operativo delle istanze. Per ulteriori informazioni, consulta Gestione degli aggiornamenti della sicurezza.

Utilizza il comando Upgrade Operating System (Aggiorna sistema operativo) per eseguire l'aggiornamento delle istanze a una nuova versione di Amazon Linux.

Upgrade Operating System (Aggiorna sistema operativo)

(Solo Linux) Aggiorna i sistemi operativi Amazon Linux delle istanze alla versione più recente. Per ulteriori informazioni, consulta AWS OpsWorks Sistemi operativi impilati.

> **Important**
>
> Dopo aver eseguito Upgrade Operating System (Aggiornamento del sistema operativo), ti consigliamo di eseguire anche Setup (Installa). Questo garantisce che i servizi vengano riavviati correttamente.

I comandi dello stack presentano le seguenti opzioni, alcune delle quali compaiono solo per determinati comandi.

Commento

(Facoltativo) Immetti i commenti personalizzati che desideri aggiungere.

Recipes to execute (Ricette da eseguire)

(Obbligatorio) Questa impostazione è disponibile solo se selezioni il comando Execute Recipes (Esegui ricette). Inserisci le ricette da eseguire utilizzando il *recipe_name* formato standard*cookbook_name*::, separate da virgole. Se specificate più ricette, AWS OpsWorks Stacks le esegue nell'ordine elencato.

Allow reboot (Consenti riavvio)

(Facoltativo) Questa impostazione è disponibile solo se selezioni il comando Upgrade Operating System (Aggiorna sistema operativo). Il valore predefinito è Yes, che indica a AWS OpsWorks Stacks di riavviare le istanze dopo l'installazione dell'aggiornamento.

Custom Chef JSON (JSON Chef personalizzato)

(Facoltativo) Scegli Advanced (Avanzate) per visualizzare questa opzione, che consente di specificare attributi JSON personalizzati da incorporare negli attributi di configurazione e distribuzione dello stack.

Istanze

(Facoltativo) Specifica le istanze su cui eseguire il comando. Tutte le istanze online sono selezionate per impostazione predefinita. Per eseguire il comando su un sottoinsieme di istanze, seleziona istanze o livelli appropriati.

> **Note**
>
> Nelle pagine Deployment (Distribuzione) e Commands (Comandi) puoi vedere le esecuzioni execute_recipes che non hai eseguito. Questo è generalmente il risultato di una modifica delle autorizzazioni, ad esempio la concessione o la rimozione di autorizzazioni SSH per un utente. Quando apporti una modifica di questo tipo, AWS OpsWorks Stacks utilizza execute_recipes per aggiornare le autorizzazioni sulle istanze.

Utilizzo di un JSON personalizzato

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Diverse azioni AWS OpsWorks Stacks consentono di specificare JSON personalizzato, che AWS OpsWorks Stacks installa sulle istanze e può essere utilizzato dalle ricette.

Puoi specificare un JSON personalizzato nelle seguenti situazioni:

- Quando crei, aggiorni o cloni uno stack.

  AWS OpsWorks Stacks installa il codice JSON personalizzato su tutte le istanze per tutti gli eventi del ciclo di vita successivi.
- Quando esegui un comando di distribuzione o dello stack.

  AWS OpsWorks Stacks passa il codice JSON personalizzato alle istanze solo per quell'evento.

Il JSON personalizzato deve essere rappresentato da, e formattato come, un oggetto JSON valido. Per esempio:

```
{
  "att1": "value1",
  "att2": "value2"
  ...
}
```

AWS OpsWorks Stacks archivia JSON personalizzato nelle seguenti posizioni:

Sulle istanze Linux:

- `/var/chef/runs/run-ID/attribs.json`

- `/var/chef/runs/run-ID/nodes/hostname.json`

Sulle istanze Windows:

- `drive:\chef\runs\run-ID\attribs.json`
- `drive:\chef\runs\run-ID\nodes\hostname.json`

> **Note**
>
> In Chef 11.10 e versioni precedenti per Linux, il JSON personalizzato è disponibile nel percorso seguente sulle istanze Linux, le istanze Windows non sono disponibili e non vi è alcun file `attribs.json`. I log vengono archiviati nella stessa cartella o directory del JSON. Per ulteriori informazioni sul JSON personalizzato in Chef 11.10 e versioni precedenti per Linux, consulta la pagina relativa alla sostituzione di attributi con JSON personalizzati e Log di Chef.
> `/var/lib/aws/opsworks/chef/hostname.json`

Nei percorsi precedenti, *run-ID* è un ID univoco che AWS OpsWorks Stacks assegna a ogni esecuzione di Chef su un'istanza ed *hostname* è il nome host dell'istanza.

Per accedere a un JSON personalizzato da ricette di Chef, usa la sintassi `node` di Chef standard.

Supponiamo che desideri definire impostazioni semplici per un'app che vuoi distribuire, ad esempio se l'app è inizialmente visibile e i colori iniziali di primo piano e di sfondo dell'app. Ipotizziamo che tu definisca queste impostazioni di app con un oggetto JSON nel modo seguente:

```
{
  "state": "visible",
  "colors": {
    "foreground": "light-blue",
    "background": "dark-gray"
  }
}
```

Per dichiarare il JSON personalizzato per uno stack:

1. Nella pagina dello stack scegliere Stack Settings (Impostazioni stack), quindi scegliere Edit (Modifica).

2. Per Custom Chef JSON (JSON Chef personalizzato), digitare l'oggetto JSON e scegliere Save (Salva).

> **Note**
>
> Puoi dichiarare un JSON personalizzato a livello di distribuzione, livello e stack. Puoi eseguire questa operazione se desideri che alcuni JSON personalizzati siano visibili solo a un singolo livello o una singola distribuzione. Oppure, ad esempio, puoi sostituire temporaneamente il JSON personalizzato dichiarato a livello di stack con il JSON personalizzato dichiarato a livello di livello. Se dichiari un JSON personalizzato su più livelli, il JSON personalizzato dichiarato a livello di distribuzione sostituisce qualsiasi JSON personalizzato dichiarato sia a livello di livello sia di stack. Il JSON personalizzato dichiarato a livello di livello sostituisce qualsiasi JSON personalizzato dichiarato solo a livello di stack.
>
> Per utilizzare la console AWS OpsWorks Stacks per specificare JSON personalizzato per una distribuzione, nella pagina Deploy App, scegli Advanced. Digitare il JSON personalizzato nella casella Custom Chef JSON (JSON Chef personalizzato) e scegliere Save (Salva).
>
> Per utilizzare la console AWS OpsWorks Stacks per specificare JSON personalizzato per un layer, nella pagina Livelli, scegli Impostazioni per il layer desiderato. Digitare il JSON personalizzato nella casella Custom JSON (JSON personalizzato) e scegliere Save (Salva).
>
> Per ulteriori informazioni, consultare Modifica della configurazione di un livello OpsWorks e Distribuzione di app.

Quando esegui un comando di distribuzione o dello stack, le ricette possono recuperare questi valori personalizzati utilizzando una sintassi `node` di Chef standard, che esegue la mappatura direttamente alla gerarchia nell'oggetto JSON personalizzato. Ad esempio, il codice ricetta seguente scrive messaggi nel log di Chef sui valori precedenti del JSON personalizzato:

```
Chef::Log.info("********** The app's initial state is '#{node['state']}' **********")
Chef::Log.info("********** The app's initial foreground color is '#{node['colors']
['foreground']}' **********")
Chef::Log.info("********** The app's initial background color is '#{node['colors']
['background']}' **********")
```

Questo approccio può essere utile per passare dati alle ricette. AWS OpsWorks Stacks aggiunge tali dati all'istanza e le ricette possono recuperarli utilizzando la sintassi standard di Chef`node`.

> **Note**
>
> Il codice JSON personalizzato è limitato a 120 KB. Se hai bisogno di maggiore capacità, ti consigliamo di archiviare alcuni dati su Amazon Simple Storage Service (Amazon S3). Le tue ricette personalizzate possono quindi utilizzare l'AWS CLI o AWS SDK per Rubyscaricare i dati dal bucket Amazon S3 sulla tua istanza.

Eliminazione di uno stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se non hai più bisogno di un stack, puoi eliminarlo. Solo gli stack vuoti possono essere eliminati; è necessario prima eliminare tutte le istanze, le app e i livelli nello stack.

Per eliminare una pila

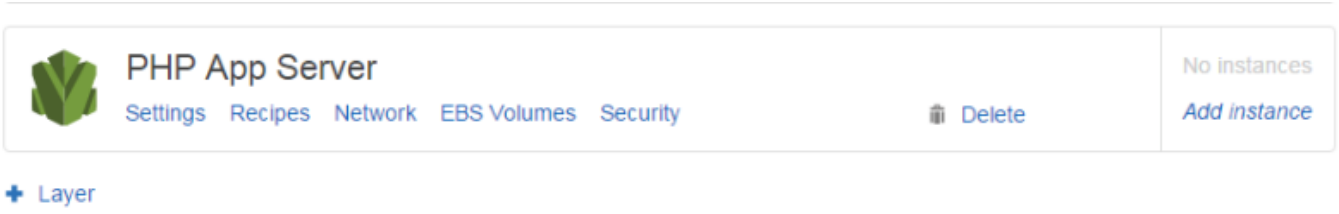
1. Nella dashboard AWS OpsWorks Stack (Stack), selezionare lo stack che si desidera chiudere ed eliminare.
2. Nel riquadro di navigazione, scegliere Instances (Istanze).
3. Nella pagina Instances (Istanze), scegliere Stop all Instances (Arresta tutte le istanze).



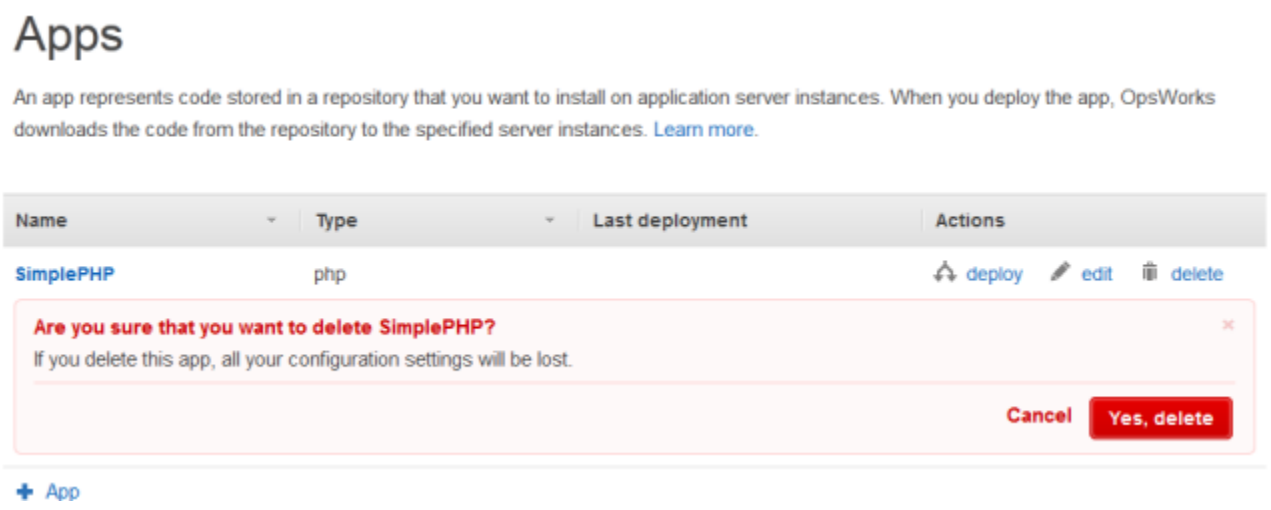
4. Dopo aver arrestato le istanze, per ogni istanza nel livello scegliere delete (eliminazione) nella colonna Actions (Operazioni). Quando viene richiesto di confermare, scegliere Yes, Delete (Sì, elimina).



5. Quando tutte le istanze sono state eliminate, nel riquadro di navigazione scegliere Layers (Livelli).
6. Nella pagina Layers (Livelli), per ogni livello nello stack scegliere delete (eliminazione). Alla richiesta di conferma, scegliere Yes, Delete (Sì, elimina).

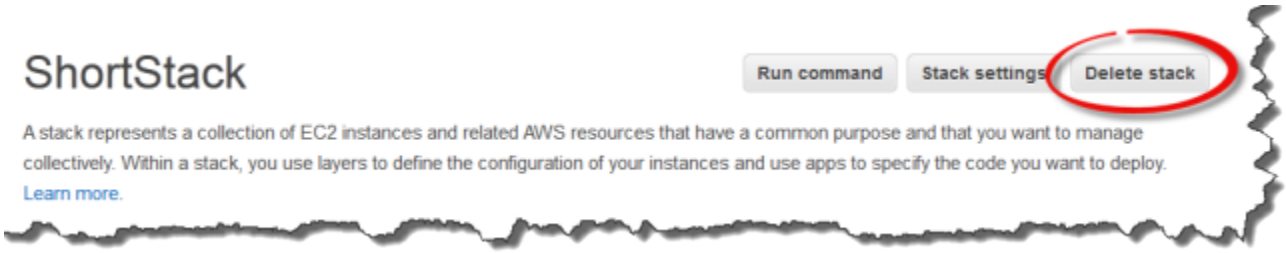


7. Quando tutti i livelli sono stati eliminati, nel riquadro di navigazione scegliere Apps (App).
8. Nella pagina Apps (App), per ogni app nello stack scegliere delete (eliminazione) nella colonna Actions (Operazioni). Alla richiesta di conferma, scegliere Yes, Delete (Sì, elimina).



9. Quando tutte le app sono state eliminate, nel riquadro di navigazione scegliere Stack.

10. Nella pagina Stack scegliere Delete stack (Elimina stack). Alla richiesta di conferma, scegliere Yes, Delete (Sì, elimina).



Eliminazione di altre AWS risorse utilizzate da uno stack

Utilizzi altre AWS risorse con AWS OpsWorks Stacks per creare e gestire i tuoi stack. Quando elimini uno stack, valuta la possibilità di eliminare anche le risorse che hanno funzionato con lo stack, se un altro stack non le utilizza e risorse esterne AWS OpsWorks a Stacks non le utilizzano. Di seguito sono riportati i motivi suggeriti per ripulire AWS le risorse esterne utilizzate con uno stack.

- AWS Le risorse esterne possono continuare ad addebitare addebiti sul tuo AWS account.
- Risorse come i bucket Amazon S3 possono contenere informazioni di identificazione personale, sensibili o riservate.

> **Important**
>
> Non eliminare queste risorse se sono utilizzate da altri stack. Tieni presente che i ruoli e i gruppi di sicurezza IAM sono globali, quindi gli stack in altre regioni potrebbero utilizzare le stesse risorse.

Di seguito sono riportate altre AWS risorse utilizzate dagli stack e collegamenti a informazioni su come eliminarle.

Ruoli di servizio e profili delle istanze

Quando crei uno stack, specifichi un ruolo IAM e un profilo di istanza che AWS OpsWorks Stacks utilizza per creare risorse consentite per tuo conto. AWS OpsWorks crea automaticamente il profilo del ruolo e dell'istanza se non scegli quelli esistenti. Il profilo di ruolo e di istanza che vengono AWS OpsWorks creati per voi sono denominati `aws-opsworks-service-role` e `aws-opsworks-ec2-role`, rispettivamente. Se nessun altro stack del tuo account utilizza il ruolo e il

profilo di istanza IAM, eliminare queste risorse è sicuro. Per informazioni sull'eliminazione di ruoli e profili delle istanze di IAM, consulta Eliminazione di ruoli o profili delle istanze nella Guida per l'utente di IAM.

Gruppi di sicurezza

In AWS OpsWorks Stacks, puoi specificare gruppi di sicurezza definiti dall'utente a livello di livello. Puoi creare gruppi di sicurezza utilizzando la EC2 console o l'API Amazon. Stack e livelli in altre regioni possono utilizzare gli stessi gruppi di sicurezza, perché questi sono globali. È possibile eliminare un gruppo di sicurezza se non è utilizzato da altre AWS risorse. Per ulteriori informazioni su come eliminare un gruppo di sicurezza, consulta Delete a security group nella Amazon EC2 User Guide.

Volumi Amazon EBS

In AWS OpsWorks Stacks, aggiungi volumi EBS a livello di livello e questi vengono collegati alle istanze del livello. Crei volumi EBS utilizzando la console di EC2 servizio Amazon o l'API, quindi li colleghi alle istanze AWS OpsWorks Stacks a livello di livello. I volumi EBS sono specifici di una zona di disponibilità. Se non si sta più usando un volume EBS in nessuno stack di una determinata regione e zona di disponibilità, è possibile eliminare il volume. Per ulteriori informazioni su come eliminare un volume Amazon EBS, consulta la sezione Eliminazione di un volume Amazon EBS nella Amazon EC2 User Guide.

Bucket Amazon Simple Storage Service (Amazon S3) S3)

In AWS OpsWorks Stacks, puoi utilizzare bucket S3 per quanto segue. I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

- Archiviazione di codice di app
- Archiviazione di libri di ricette e ricette
- CloudTrail log, se hai abilitato l'accesso a Stacks CloudTrail AWS OpsWorks
- Streams di Amazon CloudWatch Logs, se li hai abilitati in Stacks AWS OpsWorks

Indirizzi IP elastici

Se hai registrato indirizzi IP elastici con AWS OpsWorks Stacks e non hai più bisogno degli indirizzi IP elastici, puoi rilasciare l'indirizzo IP elastico.

Load balancer Elastic Load Balancing

Se non hai più bisogno di un sistema di bilanciamento del carico classico Elastic Load Balancing che hai utilizzato con i livelli dello stack, puoi eliminarlo. Per ulteriori informazioni, consulta

Eliminazione del sistema di bilanciamento del carico nella Guida per l'utente di Classic Load Balancer.

Istanze Amazon RDS (Amazon RDS)

Se hai registrato istanze di database Amazon RDS (DB) con AWS OpsWorks Stacks e non ne hai più bisogno, puoi eliminare le istanze DB. Per ulteriori informazioni su come eliminare le istanze DB, consulta Eliminazione di un'istanza DB nella Amazon RDS User Guide.

Cluster Amazon ECS (Amazon ECS ECS)

Se lo stack conteneva livelli di cluster ECS ma il cluster ECS registrato con un livello non viene più utilizzato, è possibile eliminarlo. Per ulteriori informazioni su come eliminare un cluster ECS, consulta Eliminazione di un cluster nella Amazon ECS Developer Guide.

Livelli

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ogni stack contiene uno o più livelli, ciascuno dei quali rappresenta un componente dello stack, ad esempio un sistema di bilanciamento del carico o una serie di server dell'applicazione.

Quando utilizzi i livelli AWS OpsWorks Stack, tieni presente quanto segue:

- Ogni livello in uno stack deve disporre di almeno un'istanza, anche se può averne di più.
- Ogni istanza in uno stack deve essere un membro di almeno un livello, ad eccezione delle istanze registrate.

  Non è possibile configurare direttamente un'istanza, ad eccezione di alcune impostazioni di base, ad esempio il nome host e la chiave SSH. Devi creare e configurare un livello apposito e aggiungere l'istanza al livello.

EC2 Le istanze Amazon possono facoltativamente far parte di più livelli. In tal caso, AWS OpsWorks Stacks esegue le ricette per installare e configurare pacchetti, distribuire applicazioni e così via per ciascuno dei livelli dell'istanza.

Assegnando un'istanza a più livelli, è possibile, ad esempio:

- Ridurre le spese ospitando il server del database e il sistema di bilanciamento del carico su una singola istanza.
- Utilizzare uno dei tuoi server dell'applicazione per l'amministrazione.

  Creare un livello amministrativo personalizzato e aggiungere una delle istanze del server dell'applicazione a tale livello. Le ricette del livello amministrativo configurano l'istanza del server dell'applicazione sia per eseguire attività amministrative, che per installare qualsivoglia software aggiuntivo. Le altre istanze del server dell'applicazione sono solo server dell'applicazione.

Questa sezione descrive come funzionano i livelli.

Argomenti

- OpsWorks Nozioni di base sui livelli
- Elastic Load Balancing Layer
- Amazon RDS Service Layer
- Livelli del cluster ECS
- AWS OpsWorks Stacks Layers personalizzati
- Installazioni pacchetti del sistema operativo per livello

OpsWorks Nozioni di base sui livelli

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione descrive come eseguire operazioni comuni a tutti i livelli AWS OpsWorks Stack.

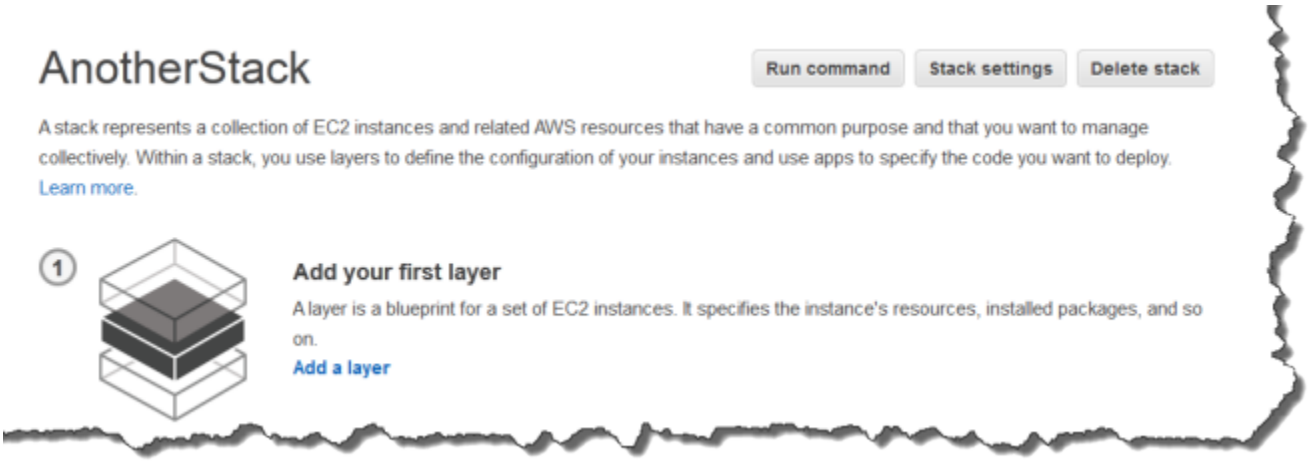
Argomenti

- Creazione di un livello OpsWorks
- Modifica della configurazione di un livello OpsWorks
- Utilizzo della diagnostica automatica per sostituire le istanze con errore
- Eliminazione di un livello OpsWorks

Creazione di un livello OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Quando crei un nuovo stack, verrà visualizzata la pagina seguente:



Per aggiungere il primo livello OpsWorks

1. Fare clic su Add a layer (Aggiungi un livello).
2. Nella pagina Add Layer (Aggiungi livello), selezionare il livello appropriato, visualizzandone così le opzioni di configurazione.

3. Configurare il livello in modo appropriato e fare clic su Add Layer (Aggiungi livello) per aggiungerlo allo stack. Le sezioni seguenti descrivono come configurare i vari livelli.

> **Note**
>
> Nella pagina Add Layer (Aggiungi livello) vengono visualizzate solo le impostazioni di configurazione più comuni per ogni livello. È possibile specificare impostazioni aggiuntive modificando il livello.

4. Aggiungere istanze al livello e avviarle.

> **Note**
>
> Se un'istanza è membro di più livelli, è necessario aggiungerla a ciascuno di essi prima di avviarla. Non è possibile aggiungere un'istanza online a un livello.

Per aggiungere ulteriori livelli, aprire la pagina Layers (Livelli) e fare clic su + Layer (+ livelli) per aprire la pagina Add Layer (Aggiungi livello).

Quando avvii un'istanza, AWS OpsWorks Stacks esegue automaticamente le ricette di installazione e distribuzione per ciascuno dei livelli dell'istanza per installare e configurare i pacchetti appropriati e distribuire le applicazioni appropriate. È possibile personalizzare il processo di installazione e configurazione di un livello in diversi modi, ad esempio assegnando ricette personalizzate agli eventi del ciclo di vita appropriati. AWS OpsWorks Stacks esegue ricette personalizzate dopo le ricette standard per ogni evento. Per ulteriori informazioni, consulta Libri di ricette e ricette.

Le seguenti sezioni specifiche dei livelli descrivono come gestire i passaggi 2 e 3 per i vari AWS OpsWorks livelli Stacks. Per ulteriori informazioni su come aggiungere le istanze, consulta Aggiunta di un'istanza a un livello.

Modifica della configurazione di un livello OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver creato un livello, alcune proprietà, come la regione AWS, non sono modificabili, ma è possibile cambiare la maggior parte della configurazione del livello in qualsiasi momento. La modifica del livello, inoltre, fornisce l'accesso a impostazioni di configurazione che non sono disponibili nella pagina Add Layer (Aggiungi livello). Le impostazioni diventano effettive non appena salvi la nuova configurazione.

Per modificare un livello OpsWorks

1. Nel riquadro di navigazione, fare clic su Layers (Livelli).
2. Nella pagina Layers (Livelli), scegliere il nome di un livello per aprire la pagina dei dettagli, che mostra la configurazione corrente.

   > **Note**
   >
   > Scegliendo uno dei nomi sotto il nome del livello si accede direttamente alla scheda associata nella pagina dei dettagli.

3. Fare clic su Modifica e quindi selezionare la scheda appropriata: General Settings (Impostazioni generali), Recipes (Ricette), Network (Rete), EBS Volumes (Volumi EBS) o Sicurezza.

Le seguenti sezioni descrivono le impostazioni delle varie schede che sono disponibili per tutti i livelli. Alcuni livelli dispongono di impostazioni aggiuntive specifiche, che vengono visualizzate nella parte superiore della pagina. Inoltre, alcune impostazioni sono disponibili solo per gli stack basati su Linux, come indicato.

Argomenti

- Impostazioni generali
- Ricette
- Rete
- Volumi EBS
- Sicurezza
- CloudWatch Registri

- Tag

Impostazioni generali

Tutti i livelli presentano le seguenti impostazioni:

Auto healing enabled (Diagnostica automatica abilitata)

Indica se la diagnostica automatica è abilitata per le istanze del livello. L'impostazione predefinita è Yes (Sì).

JSON personalizzato

I dati in formato JSON trasmessi alle tue ricette Chef per tutte le istanze in questo livello. Puoi utilizzare questa impostazione, ad esempio, per trasmettere i dati alle tue stesse ricette. Per ulteriori informazioni, consulta Utilizzo di un JSON personalizzato.

> **Note**
>
> Puoi dichiarare un JSON personalizzato a livello di distribuzione, livello e stack. Puoi eseguire questa operazione se desideri che un JSON personalizzato sia visibile sullo stack o solo in una singola distribuzione. Oppure, ad esempio, puoi sostituire temporaneamente il JSON personalizzato dichiarato a livello di livello con il JSON personalizzato dichiarato a livello di distribuzione. Se dichiari un JSON personalizzato su più livelli, il JSON personalizzato dichiarato a livello di distribuzione sostituisce qualsiasi JSON personalizzato dichiarato sia a livello di livello sia di stack. Il JSON personalizzato dichiarato a livello di livello sostituisce qualsiasi JSON personalizzato dichiarato solo a livello di stack.
>
> Per utilizzare la console AWS OpsWorks Stacks per specificare JSON personalizzato per una distribuzione, nella pagina Deploy App, scegli Avanzato. Digitare il JSON personalizzato nella casella Custom Chef JSON (JSON Chef personalizzato) e scegliere Save (Salva).
>
> Per utilizzare la console AWS OpsWorks Stacks per specificare JSON personalizzato per uno stack, nella pagina delle impostazioni dello stack, digita il codice JSON personalizzato nella casella Custom JSON, quindi scegli Salva.
>
> Per ulteriori informazioni, consultare Utilizzo di un JSON personalizzato e Distribuzione di app.

Instance shutdown timeout (Timeout di arresto istanza)

Speciifica per quanto tempo (in secondi) AWS OpsWorks Stacks attende dopo l'attivazione di un evento del ciclo di vita Shutdown prima di arrestare o terminare l'istanza Amazon. EC2 L'impostazione predefinita è 120 secondi. Lo scopo dell'impostazione è concedere alle ricette Shutdown (Arresta) dell'istanza tempo sufficiente per completare le loro attività prima di terminare l'istanza. Se le ricette Shutdown (Arresta) personalizzate richiedono più tempo, modifica l'impostazione di conseguenza. Per ulteriori informazioni sull'arresto delle istanze, consulta Arresto di un'istanza.

Le restanti impostazioni in questa scheda variano secondo il tipo di livello e sono identiche alle impostazioni nella pagina Add Layer (Aggiungi livello) del livello.

Ricette

La scheda Recipes (Ricette) include le impostazioni seguenti.

Custom Chef recipes (Ricette di Chef personalizzate)

Puoi assegnare ricette di Chef personalizzate agli eventi del ciclo di vita del livello. Per ulteriori informazioni, consulta Esecuzione di ricette.

Rete

Las scheda Network (Rete) include le impostazioni seguenti.

Elastic Load Balancing

Puoi collegare un sistema di bilanciamento del carico Elastic Load Balancing a qualsiasi livello. AWS OpsWorks Stacks quindi registra automaticamente le istanze online del layer con il sistema di bilanciamento del carico e le annulla la registrazione quando vanno offline. Se hai abilitato la funzione di drenaggio della connessione del load balancer, puoi specificare se Stacks la supporta. AWS OpsWorks Per ulteriori informazioni, consulta Elastic Load Balancing Layer.

Automatically Assign IP Addresses (Assegna automaticamente indirizzi IP)

Puoi controllare se AWS OpsWorks Stacks assegna automaticamente indirizzi IP pubblici o elastici alle istanze del layer. Ecco cosa succede quando si abilita questa opzione:

- Ad esempio, le istanze archiviate, AWS OpsWorks Stacks assegna automaticamente un indirizzo ogni volta che l'istanza viene avviata.

- Per le istanze supportate da Amazon EBS, AWS OpsWorks Stacks assegna automaticamente un indirizzo quando l'istanza viene avviata per la prima volta.
- Se un'istanza appartiene a più di un livello, AWS OpsWorks Stacks assegna automaticamente un indirizzo se hai abilitato l'assegnazione automatica per almeno uno dei livelli,

> **Note**
>
> Se abiliti l'assegnazione automatica di indirizzi IP pubblici, si applica solo alle nuove istanze. AWS OpsWorks Gli stack non possono aggiornare l'indirizzo IP pubblico per le istanze esistenti.

Se il tuo stack è in esecuzione in un VPC, hai impostazioni separate per gli indirizzi IP elastici e pubblici. La tabella seguente spiega la loro modalità di interazione:

Elastic IP addresses \ Public IP addresses	Yes	No
Yes	Instances receive an Elastic IP address when they are started for the first time, or a public IP address if an Elastic IP cannot be assigned.	Instances receive an Elastic IP address when they are started for the first time.
No	Instances receive a public IP address each time they are started.	Instances receive only a private IP address, which is not accessible from outside the VPC.

> **Note**
>
> Le istanze devono avere un modo per comunicare con il servizio AWS OpsWorks Stacks, gli archivi di pacchetti Linux e gli archivi di libri di cucina. Se non specifichi alcun indirizzo IP elastico o pubblico, il tuo VPC deve includere un componente, ad esempio un NAT, che consenta alle istanze del livello di comunicare con i siti esterni. Per ulteriori informazioni, consulta Esecuzione di uno stack in un VPC.

Se il tuo stack non è in esecuzione in un VPC, la tua unica impostazione sarà Elastic IP addresses (Indirizzi IP elastici):

- Yes (Sì): le istanze ricevono un indirizzo IP elastico quando vengono avviate per la prima volta o un indirizzo IP pubblico se non è possibile assegnare un indirizzo IP elastico.
- No: le istanze ricevono un indirizzo IP pubblico ogni volta che vengono avviate.

Volumi EBS

Las scheda EBS Volumes (Volumi EBS) include le impostazioni seguenti.

Istanze ottimizzate per EBS

Se le istanze del livello devono essere ottimizzate per Amazon Elastic Block Store (Amazon EBS). Per ulteriori informazioni, consulta Istanze ottimizzate per EBS Amazon.

Additional EBS Volumes (Volumi EBS aggiuntivi)

(Solo Linux) Puoi aggiungere o rimuovere volumi Amazon EBS dalle istanze del layer. Quando avvii un'istanza, AWS OpsWorks Stacks crea automaticamente i volumi e li collega alle istanze. Puoi utilizzare la pagina Risorse per gestire i volumi EBS dello stack. Per ulteriori informazioni, consulta Gestione delle risorse.

- Punto di montaggio: (Obbligatorio) Specificate il punto di montaggio o la directory in cui verrà montato il volume EBS.
- # Dischi — (Facoltativo) Se è stato specificato un array RAID, il numero di dischi nell'array.

  Ogni livello RAID dispone di un numero di dischi predefinito, ma è possibile selezionare un numero maggiore dall'elenco.
- Dimensione totale (GiB) — (Obbligatorio) La dimensione del volume, in GiB.

  Per una matrice RAID, questa impostazione specifica la sua dimensione totale, non le dimensioni di ciascun disco.

  La tabella riportata di seguito mostra le dimensioni minime e massime consentite per ogni tipo di volume.

Tipo di volume	Dimensione minima (GiB)	Dimensione massima (GiB)
Magnetico	1	1024
Provisioned IOPS (SSD)	4	16384
General Purpose (SSD)	1	16384
Throughput Optimized HDD	500	16384
Cold HDD	500	16384

- Tipo di volume: (facoltativo) Specificate se creare un volume SSD magnetico per scopo generico, HDD Cold o PIOPS.

  Il valore predefinito è Magnetic (Magnetico).
- Crittografato: (Facoltativo) Specificate se crittografare il contenuto del volume EBS.
- IOPS per disco — (obbligatorio per i volumi Provisioned IOPS SSD e General Purpose SSD) Se si specifica un volume Provisioned IOPS SSD o General Purpose SSD, è necessario specificare anche gli IOPS per disco.

  Per i volumi Provisioned IOPS, puoi specificare la frequenza IOPS al momento della creazione del volume. Il rapporto tra Provisioned IOPS e le dimensioni richieste relative al volume può essere massimo pari a 30 (in altre parole, un volume con 3.000 IOPS deve essere almeno di 100 GB). I tipi di volume General Purpose (SSD) hanno uno IOPS di riferimento per quanto riguarda le dimensioni del volume x 3 con un massimo di 10.000 IOPS e possono eseguire fino a 3.000 IOPS per 30 minuti.

Quando aggiungi o rimuovi volumi da un livello, tieni presente quanto segue:

- Se aggiungi un volume, ogni nuova istanza ottiene il nuovo volume, ma AWS OpsWorks Stacks non aggiorna le istanze esistenti.
- L'eliminazione di un volume si applica solo alle nuove istanze; le istanze esistenti mantengono i propri volumi.

Specificare un punto di montaggio

Puoi specificare qualsiasi punto di montaggio desideri. Tuttavia, tieni presente che alcuni punti di montaggio sono riservati all'uso da parte di AWS OpsWorks Stacks o Amazon EC2 e non devono essere utilizzati per i volumi Amazon EBS. Non utilizzare le cartelle di sistema Linux tipiche come `/home` o `/etc`.

I seguenti punti di montaggio sono riservati all'uso da parte AWS OpsWorks di Stacks.

- `/srv/www`
- `/var/log/apache2` (Ubuntu)
- `/var/log/httpd` (Amazon Linux)
- `/var/log/mysql`
- `/var/www`

Quando un'istanza si avvia o riavvia, autofs (un daemon di montaggio automatico) usa punti di montaggio di dispositivi temporanei come `/media/ephemeral0` per vincolare montaggi. Questa operazione viene eseguita prima del montaggio dei volumi Amazon EBS. Per garantire che il punto di montaggio del volume Amazon EBS non sia in conflitto con autofs, non specificare un punto di montaggio temporaneo del dispositivo. I possibili punti di montaggio temporanei del dispositivo dipendono dal particolare tipo di istanza e dal fatto che sia supportata da Instance Store o da Amazon EBS. Per evitare un conflitto con autofs, procedi nel seguente modo:

- Verifica i punti di montaggio di dispositivi temporanei per il tipo di istanza specifico e l'archivio di backup che intendi utilizzare.
- Tieni presente che un punto di montaggio che funziona per un'istanza supportata da Instance Store potrebbe entrare in conflitto con autofs se passi a un'istanza supportata da Amazon EBS o viceversa.

> **Note**
>
> Se intendi modificare la mappatura dei dispositivi a blocchi dell'instance store, puoi creare un'AMI personalizzata. Per ulteriori informazioni, consulta Amazon EC2 Instance Store. Per ulteriori informazioni su come creare un'AMI personalizzata per AWS OpsWorks Stack, consultaUtilizzo di Custom AMIs.

Di seguito è riportato un esempio di come utilizzare una ricetta personalizzata per assicurarsi che il punto di montaggio di un volume non entri in conflitto con autofs. Puoi adattarlo in base alle esigenze per il tuo caso d'uso specifico.

Per evitare un punto di montaggio conflittuale

1. Assegna un volume Amazon EBS al layer desiderato, ma utilizza un punto di montaggio come `/mnt/workspace` quello che non entrerà mai in conflitto con autofs.
2. Implementa la seguente ricetta personalizzata, che crea una directory applicativa sul volume Amazon EBS e da `/srv/www/` cui si collega. Per ulteriori informazioni su come implementare le ricette personalizzate, consulta Libri di ricette e ricette e Personalizzazione degli stack AWS OpsWorks.

```
mount_point = node['ebs']['raids']['/dev/md0']['mount_point'] rescue nil
```

```
if mount_point
  node[:deploy].each do |application, deploy|
    directory "#{mount_point}/#{application}" do
      owner deploy[:user]
      group deploy[:group]
      mode 0770
      recursive true
    end

    link "/srv/www/#{application}" do
      to "#{mount_point}/#{application}"
    end
  end
end
```

3. Aggiungere una riga `depends 'deploy'` al file `metadata.rb` del libro di ricette personalizzato.
4. Assegnare questa ricetta all'evento Setup (Imposta) del livello.

Sicurezza

La scheda Sicurezza include le impostazioni seguenti.

Gruppi di sicurezza

Un livello deve avere almeno un gruppo di sicurezza associato. Specifichi come associare i gruppi di sicurezza quando crei o aggiorni uno stack. AWS OpsWorks Stacks fornisce un set standard di gruppi di sicurezza integrati.

- L'opzione predefinita prevede che AWS OpsWorks Stack associa automaticamente il gruppo di sicurezza integrato appropriato a ogni livello.
- Puoi anche scegliere di non associare automaticamente i gruppi di sicurezza integrati ma di associare un gruppo di sicurezza personalizzato a ciascun livello quando crei quest'ultimo.

Per ulteriori informazioni sui gruppi di sicurezza, consulta Utilizzo dei gruppi di sicurezza.

Dopo che il livello è stato creato, puoi utilizzare Security Groups (Gruppi di sicurezza) per aggiungere ulteriori gruppi di sicurezza al livello selezionandoli dall'elenco Custom security groups (Gruppi di sicurezza personalizzati). Dopo aver aggiunto un gruppo di sicurezza a un livello, AWS OpsWorks Stacks lo aggiunge a tutte le nuove istanze. (Tieni presente che le istanze di instance-

store che vengono riavviate verranno visualizzate come nuove istanze, quindi avranno anche i nuovi gruppi di sicurezza.) AWS OpsWorks Stacks non aggiunge gruppi di sicurezza alle istanze online.

Puoi eliminare i gruppi di sicurezza esistenti facendo clic sulla x come segue:

- Se hai scelto che AWS OpsWorks Stacks associ automaticamente i gruppi di sicurezza integrati, puoi eliminare i gruppi di sicurezza personalizzati aggiunti in precedenza facendo clic sulla x, ma non puoi eliminare il gruppo integrato.
- Se hai scelto di non associare automaticamente i gruppi di sicurezza integrati, puoi eliminare i gruppi di sicurezza esistenti, tra cui quello originale, purché il livello conservi almeno un gruppo.

Dopo aver rimosso un gruppo di sicurezza da un livello, AWS OpsWorks Stacks non lo aggiunge a nessuna istanza nuova o riavviata. AWS OpsWorks Stacks non rimuove i gruppi di sicurezza dalle istanze online.

> **Note**
>
> Se lo stack è in esecuzione in un VPC, puoi aggiungere o rimuovere un gruppo di sicurezza per un'istanza online utilizzando la console EC2 Amazon, l'API o la CLI. Tuttavia, questo gruppo di sicurezza non sarà visibile nella console Stacks. AWS OpsWorks Se desideri rimuovere il gruppo di sicurezza, devi utilizzare anche Amazon EC2. Per ulteriori informazioni, consulta l'articolo relativo ai gruppi di sicurezza.

Tieni presente quanto segue:

- Non è possibile limitare le impostazioni di accesso della porta di un gruppo di sicurezza integrato aggiungendo un gruppo di sicurezza più restrittivo. Quando sono presenti più gruppi di sicurezza, Amazon EC2 utilizza le impostazioni più permissive.
- Si raccomanda di non modificare la configurazione di un gruppo di sicurezza integrato. Quando crei uno stack, AWS OpsWorks Stacks sovrascrive le configurazioni dei gruppi di sicurezza integrati, quindi tutte le modifiche apportate andranno perse alla successiva creazione di uno stack.

Se realizzi di aver bisogno di impostazioni del gruppo di sicurezza più restrittive per uno o più livelli, segui questa procedura:

1. Crea gruppi di sicurezza personalizzati con le impostazioni appropriate e aggiungili ai livelli pertinenti.

Ogni livello nel tuo stack deve disporre di almeno un gruppo di sicurezza, in aggiunta al gruppo integrato, anche se un solo livello richiede le impostazioni personalizzate.

2. Modifica la configurazione dello stack e imposta l'impostazione Usa gruppi di sicurezza su No. OpsWorks

   AWS OpsWorks Stacks rimuove automaticamente il gruppo di sicurezza integrato da ogni livello.

Per ulteriori informazioni sui gruppi di sicurezza, consulta Amazon EC2 Security Groups.

EC2 Profilo dell'istanza

Puoi modificare il EC2 profilo per le istanze del layer. Per ulteriori informazioni, consulta Specificazione delle autorizzazioni per le app in esecuzione su istanze EC2 .

CloudWatch Registri

La scheda CloudWatch Logs consente di abilitare o disabilitare Amazon CloudWatch Logs. CloudWatch L'integrazione dei log funziona con gli stack basati su Chef 11.10 e Chef 12 Linux. Per ulteriori informazioni sull'attivazione dell'integrazione con CloudWatch Logs e sulla specificazione dei log che si desidera gestire nella console Logs, vedere. CloudWatch Utilizzo di Amazon CloudWatch Logs con Stacks AWS OpsWorks

Tag

La scheda Tag ti consente di applicare tag per l'allocazione dei costi al tuo livello. Dopo aver aggiunto i tag, puoi attivarli nella console. AWS Billing and Cost Management Quando crei un tag, lo stai applicando a ogni risorsa all'interno della struttura con tag. Ad esempio, se applichi un tag a un layer, lo applichi a ogni istanza, volume Amazon EBS o sistema di bilanciamento del carico Elastic Load Balancing nel layer. Per ulteriori informazioni su come attivare i tag e utilizzarli per tracciare e gestire i costi delle risorse AWS OpsWorks Stacks, consulta Utilizzo dei tag di allocazione dei costi e Attivazione dei tag di allocazione dei costi definiti dall'utente nella Guida per l'utente di Billing and Cost Management. Per ulteriori informazioni sul tagging in AWS OpsWorks Stacks, consulta Tag.

Utilizzo della diagnostica automatica per sostituire le istanze con errore

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ogni istanza ha un agente AWS OpsWorks Stacks che comunica regolarmente con il servizio. AWS OpsWorks Stacks utilizza tale comunicazione per monitorare lo stato dell'istanza. Se un agente non comunica con il servizio per più di circa cinque minuti, AWS OpsWorks Stacks considera l'istanza fallita.

La diagnostica automatica è impostata a livello di livello; puoi modificare questa impostazione cambiando le impostazioni del livello, come illustrato nella screenshot seguente.



Note

Un'istanza può essere membro di più livelli. Se uno di questi livelli ha la correzione automatica disattivata, AWS OpsWorks Stacks non corregge l'istanza in caso di errore.

Se un livello ha la correzione automatica abilitata, l'impostazione predefinita,AWS OpsWorks Stacks sostituisce automaticamente le istanze fallite del livello come segue:

Istanza supportata da instance store

1. Arresta l' EC2 istanza Amazon e verifica che sia stata chiusa.

2. Elimina i dati del volume root.
3. Crea una nuova EC2 istanza Amazon con lo stesso nome host, configurazione e appartenenza al livello.
4. Ricollega tutti i volumi Amazon EBS, inclusi i volumi collegati dopo l'avvio originario della vecchia istanza.
5. Assegna un nuovo indirizzo IP pubblico e privato.
6. Se la vecchia istanza era associata a un indirizzo IP elastico, associarla alla nuova istanza con lo stesso indirizzo IP.

Istanza supportata da Amazon EBS

1. Arresta l' EC2 istanza Amazon e verifica che sia stata interrotta.
2. Avvia l' EC2 istanza.

Dopo che l'istanza con riparazione automatica è tornata online, AWS OpsWorks Stacks attiva un evento Configure lifecycle su tutte le istanze dello stack. Gli attributi di configurazione e distribuzione dello stack associati includono indirizzi IP pubblici e privati dell'istanza. Le ricette di configurazione (Configura) personalizzate possono ottenere i nuovi indirizzi IP dall'oggetto nodo.

Se specifichi un volume Amazon EBS per le istanze di un layer, AWS OpsWorks Stacks crea un nuovo volume e lo collega a ciascuna istanza all'avvio dell'istanza. Se successivamente vuoi scollegare il volume da un'istanza, utilizza la pagina relativa alle risorse.

Quando AWS OpsWorks Stacks cura automaticamente una delle istanze di un livello, gestisce i volumi nel modo seguente:

- Se il volume è stato collegato all'istanza quando l'istanza ha avuto esito negativo, il volume e i relativi dati vengono salvati e AWS OpsWorks Stacks lo collega alla nuova istanza.
- Se il volume non era collegato all'istanza nel momento dell'errore di quest'ultima, AWS OpsWorks Stacks crea un nuovo volume vuoto con la configurazione specificata dal livello e lo collega alla nuova istanza.

La diagnostica automatica è abilitata per impostazione predefinita su tutti i livelli, ma è possibile modificare le impostazioni generali del livello per disabilitarla.

> **Important**
>
> Se la diagnostica automatica è abilitata, assicurati di eseguire le operazioni seguenti:

- Usa solo la console AWS OpsWorks Stacks, la CLI o l'API per interrompere le istanze.

  Se interrompi un'istanza in altro modo, ad esempio utilizzando la EC2 console Amazon, AWS OpsWorks Stacks considera l'istanza come fallita e la ripara automaticamente.
- Usa i volumi Amazon EBS per archiviare tutti i dati che non vuoi perdere se l'istanza viene riparata automaticamente.

  La correzione automatica interrompe la vecchia EC2 istanza Amazon, che distrugge tutti i dati che non sono archiviati su un volume Amazon EBS. I volumi Amazon EBS vengono ricollegati alla nuova istanza, che conserva tutti i dati archiviati.

Eliminazione di un livello OpsWorks

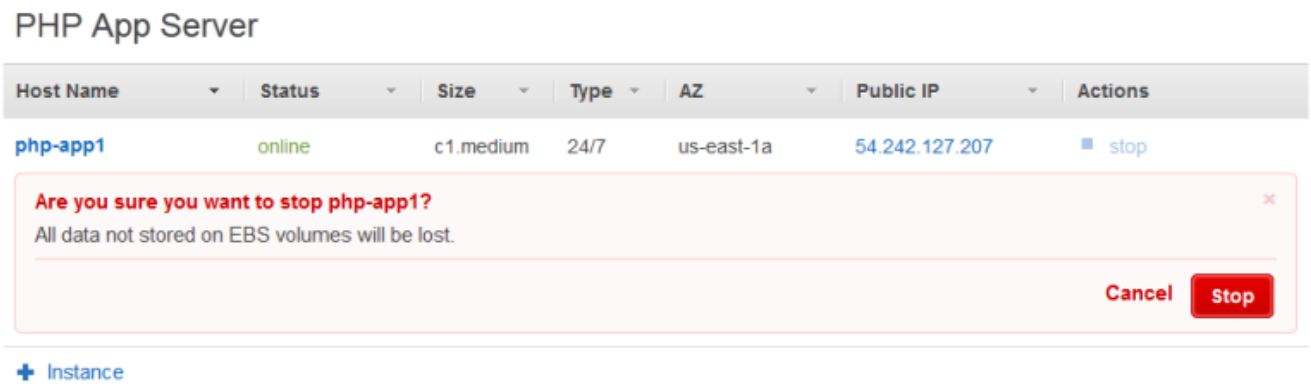
Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

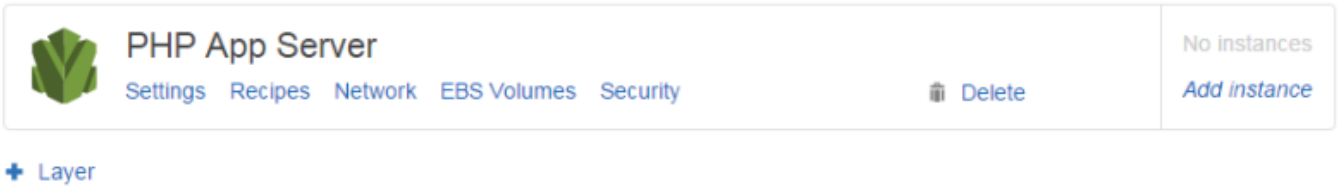
Se non hai più bisogno di un livello AWS OpsWorks Stack, puoi eliminarlo dallo stack.

Per eliminare un livello OpsWorks

1. Nel riquadro di navigazione fare clic su Instances (Istanze).
2. Nella pagina Instances (Istanze), sotto il nome del livello che si desidera eliminare, fare clic su stop (arresta) nella colonna Operazioni per ogni istanza.



3. Dopo l'arresto di ogni istanza, fare clic su elimina per rimuoverla dal livello.
4. Nel riquadro di navigazione, fare clic su Layers (Livelli).
5. Nella pagina Layers (Livelli), scegliere Elimina.



Elastic Load Balancing Layer

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Elastic Load Balancing funziona in modo leggermente diverso rispetto a un livello AWS OpsWorks Stacks. Invece di creare un layer e aggiungervi istanze, utilizzate la console o l'API Elastic Load Balancing per creare un load balancer e quindi collegarlo a un layer esistente. Oltre a distribuire il traffico verso le istanze del layer, Elastic Load Balancing esegue le seguenti operazioni:

- Rileva le EC2 istanze Amazon non integre e re-instrada il traffico alle istanze integre rimanenti finché quelle non integre non vengono ripristinate.

- Ridimensiona automaticamente la capacità di gestione delle richieste in risposta al traffico in entrata.
- Se si abilita Connection Draining, il sistema di bilanciamento del carico invia nuove richieste alle istanze non integre o la cui registrazione sta per essere annullata, ma mantiene la connessione attiva, fino a un valore di timeout specificato, per consentire all'istanza di completare qualsiasi richiesta in transito.

Dopo aver collegato un sistema di bilanciamento del carico a un layer, Stacks esegue le seguenti operazioni: AWS OpsWorks

- Annulla la registrazione di qualsiasi istanza registrata attualmente.
- Registra automaticamente le istanze del livello quando sono online e annulla la registrazione delle istanze quando sono offline, tra cui istanze basate sul carico e sul tempo.
- Avvia automaticamente le richieste di routing per le istanze registrate nelle zone di disponibilità.

Se hai abilitato la funzione di drenaggio della connessione del load balancer, puoi specificare se AWS OpsWorks Stacks la supporta. Se abiliti il supporto per il drenaggio della connessione (impostazione predefinita), dopo la chiusura di un'istanza, AWS OpsWorks Stacks esegue le seguenti operazioni:

- Annulla la registrazione dell'istanza dal sistema di bilanciamento del carico.

  Il sistema di bilanciamento del carico smette di inviare nuove richieste e avvia il Connection Draining.
- Ritarda l'attivazione di un evento di arresto del ciclo di vita fino a che il sistema di bilanciamento del carico ha completato il Connection Draining.

Se non abiliti il supporto per il drenaggio della connessione, AWS OpsWorks Stacks attiva l'evento Shutdown non appena l'istanza viene chiusa, anche se l'istanza è ancora connessa al load balancer.

Per utilizzare Elastic Load Balancing con uno stack, devi prima creare uno o più sistemi di bilanciamento del carico nella stessa regione utilizzando la console Elastic Load Balancing, la CLI o l'API. Ricordarsi quanto segue:

- È possibile collegare solo un sistema di bilanciamento del carico a un livello.
- Ogni sistema di bilanciamento del carico è in grado di gestire un solo livello.

- AWS OpsWorks Stacks non supporta Application Load Balancer. Puoi usare Classic Load Balancer solo con AWS OpsWorks Stacks.

Ciò significa che è necessario creare un sistema di bilanciamento del carico Elastic Load Balancing separato per ogni livello di ogni stack che si desidera bilanciare e utilizzarlo solo per tale scopo. Una pratica consigliata consiste nell'assegnare un nome distintivo a ciascun sistema di bilanciamento del carico Elastic Load Balancing che intendi utilizzare AWS OpsWorks con gli stack, ad esempio MyStack 1 RailsLayer - -ELB, per evitare di utilizzare un load balancer per più di uno scopo.

> **Important**
>
> Ti consigliamo di creare nuovi sistemi di bilanciamento del carico Elastic Load Balancing per i livelli Stacks. AWS OpsWorks Se scegli di utilizzare un sistema di bilanciamento del carico Elastic Load Balancing esistente, devi prima confermare che non viene utilizzato per altri scopi e che non ha istanze collegate. Dopo aver collegato il load balancer al layer, OpsWorks rimuove tutte le istanze esistenti e configura il load balancer per gestire solo le istanze del layer. Sebbene sia tecnicamente possibile utilizzare la console o l'API Elastic Load Balancing per modificare la configurazione di un load balancer dopo averlo collegato a un layer, non dovresti farlo; le modifiche non saranno permanenti.

Per collegare un sistema di bilanciamento del carico Elastic Load Balancing a un livello

1. Se non l'hai ancora fatto, usa la console Elastic Load Balancing, l'API o la CLI per creare un load balancer nella regione dello stack. Quando si crea un sistema di bilanciamento del carico, procedere nel seguente modo:

   - Assicurarsi di specificare un percorso ping del controllo dello stato adatto per l'applicazione.

     Il percorso ping predefinito è `/index.html`, perciò se l'applicazione root non include `index.html`, è necessario specificare un percorso ping appropriato o il controllo dello stato darà errori.
   - Se si desidera utilizzare Connection Draining, assicurarsi che la funzionalità sia attiva e che abbia un valore di timeout appropriato.

   Per ulteriori informazioni, consulta Elastic Load Balancing.

2. Creazione del livello che si desidera equilibrare o modificare le impostazioni di rete di un livello esistente.

   > **Note**
   >
   > Non è possibile collegare un sistema di bilanciamento del carico quando si crea un livello personalizzato. È necessario modificare le impostazioni del livello.

3. In Elastic Load Balancing, seleziona il load balancer che desideri collegare al layer e specifica se desideri che AWS OpsWorks Stacks supporti il drenaggio delle connessioni.

Dopo aver collegato un load balancer a un layer, AWS OpsWorks Stacks attiva un evento Configure lifecycle sulle istanze dello stack per notificare loro la modifica. AWS OpsWorks Stacks attiva anche un evento Configure quando scolleghi un load balancer.

> **Note**
>
> Dopo l'avvio di un'istanza, AWS OpsWorks Stacks esegue le ricette Setup and Deploy, che installano pacchetti e distribuiscono applicazioni. Al termine di tali ricette, l'istanza è nello stato online e AWS OpsWorks Stacks la registra con Elastic Load Balancing. AWS OpsWorks Stacks attiva anche un evento Configure dopo che l'istanza è online. Ciò significa che la registrazione di Elastic Load Balancing e le ricette di configurazione potrebbero essere eseguite contemporaneamente e l'istanza potrebbe essere registrata prima del termine delle ricette di configurazione. Per garantire che una ricetta finisca prima che un'istanza venga registrata con Elastic Load Balancing, è necessario aggiungere la ricetta agli eventi del ciclo di vita Setup o Deploy del layer. Per ulteriori informazioni, consulta Esecuzione di ricette.

Talvolta è utile rimuovere un'istanza da un sistema di bilanciamento del carico. Ad esempio, quando si aggiorna un'app, si consiglia di distribuire l'app in una singola istanza e verificare che l'app funzioni correttamente prima di distribuirla in ogni istanza. Quell'istanza viene in genere rimossa dal sistema di bilanciamento del carico, in modo che non riceva le richieste dagli utenti finché non si è verificato l'aggiornamento.

È necessario utilizzare la console o l'API Elastic Load Balancing per rimuovere temporaneamente un'istanza online da un sistema di bilanciamento del carico. La seguente sezione descrive come utilizzare la console.

Per rimuovere temporaneamente un'istanza da un sistema di bilanciamento del carico

1. Apri la EC2 console Amazon e scegli Load Balancers.
2. Scegliere il sistema di bilanciamento del carico appropriato e aprire la scheda Instances (Istanze).
3. Selezionare Remove from Load Balancer (Rimuovi da Sistema di bilanciamento del carico) nella colonna dell'istanza Actions (Operazioni).
4. Al termine, selezionare Edit Instances (Modifica istanze) e tornare l'istanza al sistema di bilanciamento del carico.

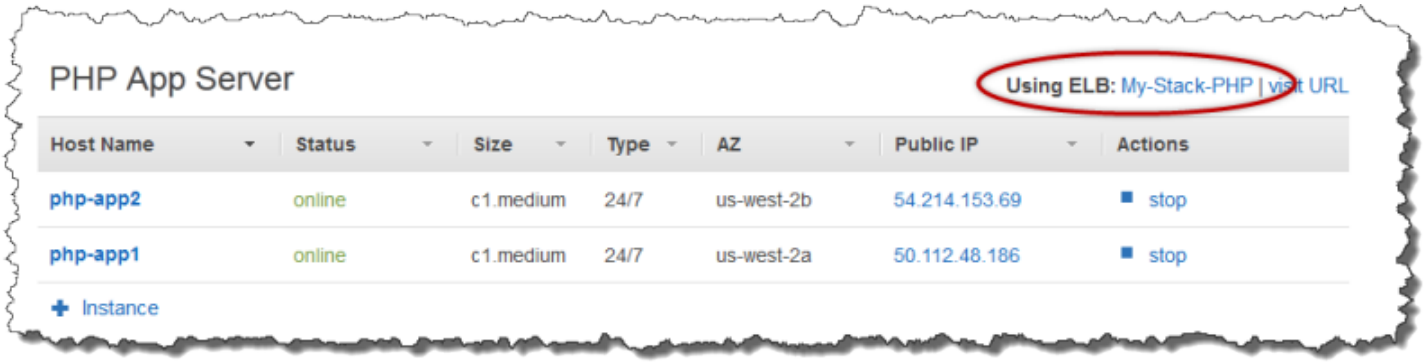
> **Important**
>
> Se utilizzi la console o l'API Elastic Load Balancing per rimuovere un'istanza da un load balancer, devi anche utilizzare Elastic Load Balancing per ripristinarla. AWS OpsWorks Stacks non è a conoscenza delle operazioni che esegui con altre console di servizio e non restituirà l'istanza al sistema di bilanciamento del carico al posto tuo. APIs A volte, AWS OpsWorks Stacks può aggiungere nuovamente l'istanza all'ELB, ma questo non è un comportamento garantito e non si verifica in tutti i casi.

È possibile collegare più sistemi di bilanciamento del carico per un determinato set di istanze come segue:

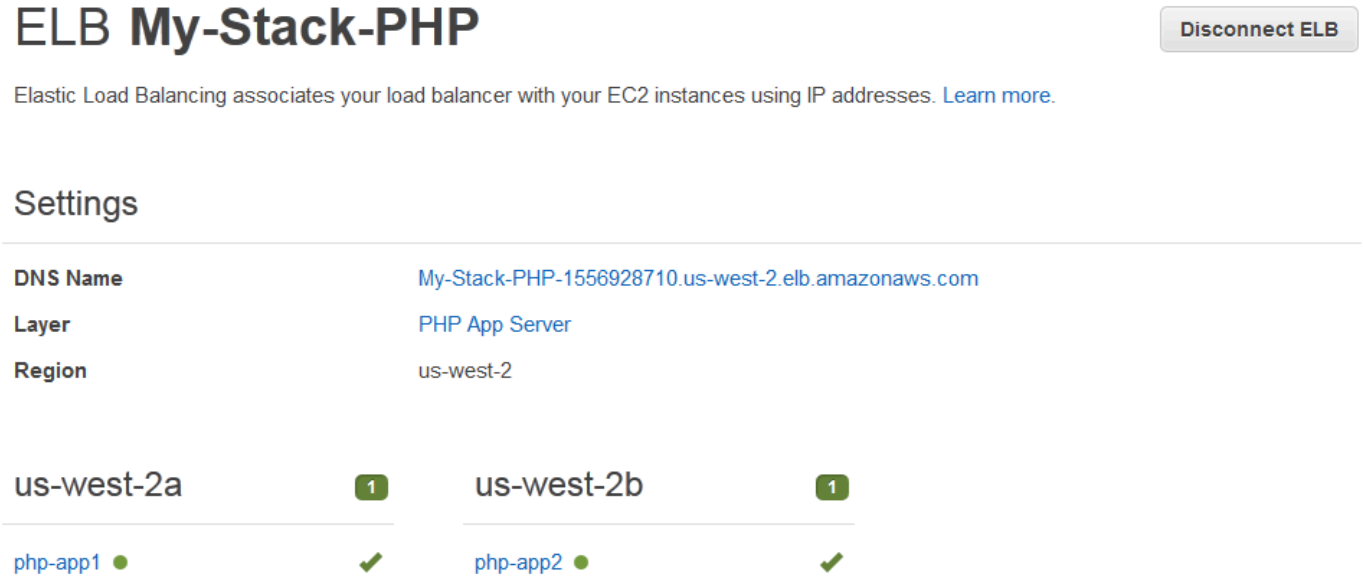
Per collegare più sistemi di bilanciamento del carico

1. Utilizza la console Elastic Load Balancing, l'API o la CLI per creare un set di sistemi di bilanciamento del carico.
2. Creare un livello personalizzato per ogni sistema di bilanciamento del carico e collegarvi uno dei sistemi di bilanciamento del carico. Non è necessario implementare ricette personalizzate per questi livelli; un livello personalizzato predefinito è sufficiente.
3. Aggiungere il set di istanze per ciascun livello personalizzato.

È possibile esaminare le proprietà di un sistema di bilanciamento del carico, visitando la pagina delle istanze e facendo clic sul nome appropriato del sistema di bilanciamento del carico.



La pagina ELB mostra le proprietà di base del sistema di bilanciamento del carico, tra cui il nome DNS e lo stato di integrità delle istanze associate. Se lo stack è in esecuzione in un VPC, la pagina mostra le sottoreti anziché le zone di disponibilità. Un segno di spunta verde indica un'istanza sana. È possibile fare clic sul nome per connettersi a un server, tramite il sistema di bilanciamento del carico.



Amazon RDS Service Layer

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Un livello di servizio Amazon RDS rappresenta un'istanza Amazon RDS. Il layer può rappresentare solo istanze Amazon RDS esistenti, che devi creare separatamente utilizzando la console o l'API di Amazon RDS.

La procedura di base per incorporare un livello di servizio Amazon RDS nello stack è la seguente:

1. Usa la console, l'API o la CLI di Amazon RDS per creare un'istanza.

   Assicurarsi di registrare l'ID dell'istanza, il nome utente master, la password master e il nome del database.
2. Per aggiungere un layer Amazon RDS al tuo stack, registra l'istanza Amazon RDS con lo stack.
3. Collega il layer a un'app, che aggiunge le informazioni di connessione dell'istanza Amazon RDS agli `deploy`attributi dell'app.
4. Utilizza i file specifici della lingua o le informazioni negli `deploy` attributi per connettere l'applicazione all'istanza Amazon RDS.

   Per ulteriori informazioni su come collegare un'applicazione a un server di database, consulta the section called “Connessione a un database”

> **Warning**
>
> Assicurarsi che i caratteri nel nome utente e nella password master dell'istanza siano compatibili con il server di applicazione. Ad esempio, con il livello Java App Server, l'inclusione & in una delle due stringhe causa un errore di analisi XML che impedisce l'avvio del server Tomcat.

Argomenti

- Specificazione dei gruppi di sicurezza
- Registrazione di un'istanza Amazon RDS con uno stack
- Associazione dei livelli di servizio Amazon RDS alle app
- Rimuovere un Amazon RDS Service Layer da uno stack

Specificazione dei gruppi di sicurezza

Per utilizzare un'istanza Amazon RDS con AWS OpsWorks Stacks, il database o i gruppi di sicurezza VPC devono consentire l'accesso dagli indirizzi IP appropriati. Per l'uso di produzione, un gruppo di sicurezza in genere limita l'accesso solo gli indirizzi IP che devono accedere al database. In genere include gli indirizzi dei sistemi utilizzati per gestire il database e le istanze AWS OpsWorks Stacks che devono accedere al database. AWS OpsWorks Stacks crea automaticamente un gruppo EC2 di sicurezza Amazon per ogni tipo di livello quando crei il tuo primo stack in una regione. Un modo semplice per fornire l'accesso alle istanze AWS OpsWorks Stacks consiste nell'assegnare i gruppi di sicurezza AWS OpsWorks Stacks appropriati all'istanza Amazon RDS o al VPC.

Per specificare i gruppi di sicurezza per un'istanza Amazon RDS esistente

1. Aprire la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Fai clic su Istanze nel riquadro di navigazione e seleziona l'istanza Amazon RDS appropriata. Fare clic su Instance Actions (Azioni istanza), Modify (Modifica).
3. Selezionare i seguenti gruppi di sicurezza dall'elenco Security Group (Gruppo di sicurezza) poi fare clic su Continue (Continua) e Modify DB Instance (Modifica istanza database) per aggiornare l'istanza.

    - Il gruppo di sicurezza AWS- OpsWorks -DB-master-Server () ***security_group_id***.
    - Il gruppo di sicurezza per il livello del server della app le cui istanze si collegheranno al database. Il nome del gruppo include il nome del livello. Ad esempio, per fornire l'accesso al database alle istanze di PHP App Server, specifica il gruppo AWS- OpsWorks -PHP-App-Server.

Se stai creando una nuova istanza Amazon RDS, puoi specificare i gruppi di sicurezza AWS OpsWorks Stacks appropriati nella pagina Configure Advanced Settings della procedura guidata Launch DB Instance. Per una descrizione su come utilizzare questa procedura guidata, consulta Creazione di un'istanza database MySQL e connessione a un database su un'istanza database MySQL.

Per informazioni su come specificare i gruppi di sicurezza VPC, consulta Gruppi di sicurezza per il VPC.

Registrazione di un'istanza Amazon RDS con uno stack

Per aggiungere un livello di servizio Amazon RDS in uno stack, devi registrare un'istanza nello stack.

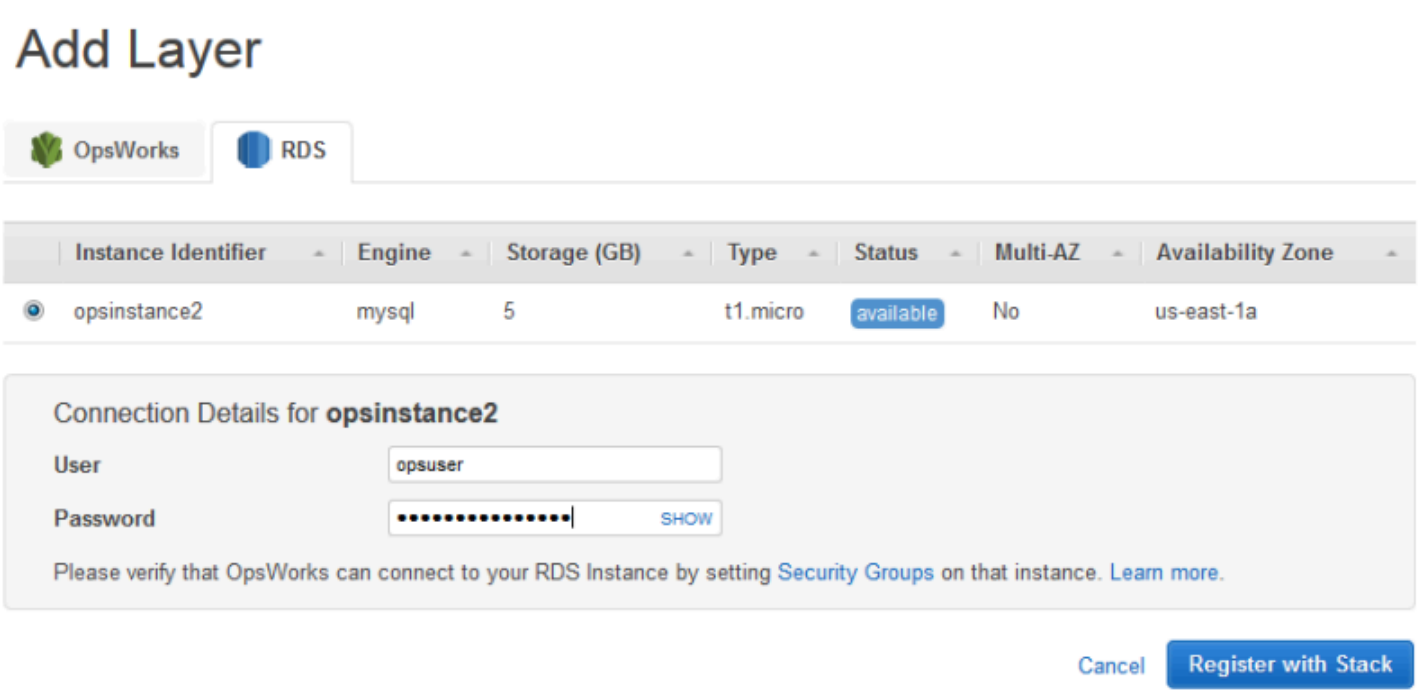
Per registrare un'istanza Amazon RDS su uno stack

1. Nella console AWS OpsWorks Stacks, fai clic su Layer nel pannello di navigazione, fai clic su + Layer o Aggiungi un layer per aprire la pagina Aggiungi livello, quindi fai clic sulla scheda RDS.
2. Se necessario, aggiornare il ruolo di servizio dello stack, come descritto in Aggiornamento del ruolo di servizio dello stack.
3. Fai clic sulla scheda RDS per elencare le istanze Amazon RDS disponibili.

> **Note**
>
> Se il tuo account non dispone di istanze Amazon RDS, puoi crearne una facendo clic su Aggiungi un'istanza RDS nella scheda RDS, che ti porta alla console Amazon RDS e avvia la procedura guidata Launch a DB Instance. Puoi anche accedere direttamente alla console Amazon RDS e fare clic su Avvia un'istanza DB oppure utilizzare l'API o la CLI di Amazon RDS. Per ulteriori informazioni su come creare un'istanza Amazon RDS, consulta Nozioni di base su Amazon RDS.

4. Selezionare l'istanza appropriata, impostare User (Utente) e Password (Password) sui valori utente e password appropriati e fare clic su Register to Stack (Registrati allo stack).

> **Important**
>
> Devi assicurarti che l'utente e la password utilizzati per registrare l'istanza Amazon RDS corrispondano a un utente e una password validi. In caso contrario, le tue applicazioni non saranno in grado di connettersi all'istanza. Tuttavia, è possibile modificare il livello per fornire nome utente e password validi valori e quindi ridistribuire l'app.



Quando aggiungi un livello di servizio Amazon RDS a uno stack, AWS OpsWorks Stacks gli assegna un ID e aggiunge la configurazione Amazon RDS associata all'attributo di configurazione e distribuzione dello stack. `[:opsworks][:stack]`

> **Note**
>
> Se modifichi la password di un'istanza Amazon RDS registrata, devi aggiornare manualmente la password in AWS OpsWorks Stacks e quindi ridistribuire le app per aggiornare la configurazione dello stack e gli attributi di distribuzione sulle istanze dello stack.

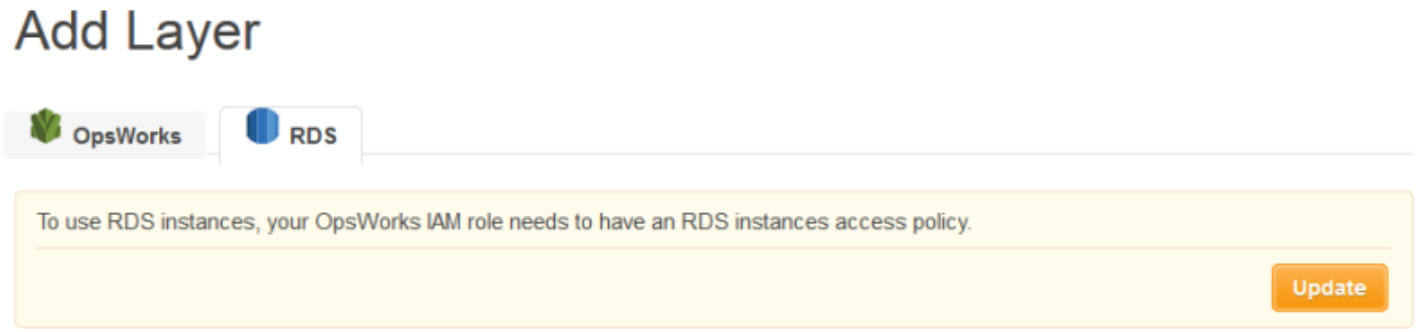
Argomenti

- Aggiornamento del ruolo di servizio dello stack

Aggiornamento del ruolo di servizio dello stack

Ogni stack ha un ruolo di servizio IAM che specifica quali azioni AWS OpsWorks Stacks può eseguire per tuo conto con altri servizi AWS. Per registrare un'istanza Amazon RDS con uno stack, il relativo ruolo di servizio deve concedere a AWS OpsWorks Stacks le autorizzazioni per accedere ad Amazon RDS.

La prima volta che aggiungi un livello di servizio Amazon RDS a uno dei tuoi stack, il ruolo di servizio potrebbe non avere le autorizzazioni richieste. In questo caso, quando si fa clic sulla scheda RDS nella pagina Add Layer (Aggiungi livello), verrà visualizzato quanto segue.



Fai clic su Aggiorna per fare in modo che AWS OpsWorks Stacks aggiorni la politica del ruolo di servizio come segue.

```
{"Statement": [{"Action": ["ec2:*", "iam:PassRole",
                       "cloudwatch:GetMetricStatistics",
                       "elasticloadbalancing:*",
                       "rds:*"],
            "Effect": "Allow",
            "Resource": ["*"] }]
}
```

> **Note**
>
> Bisogna eseguire l'aggiornamento solo una volta. Il ruolo aggiornato viene quindi automaticamente utilizzato da tutti gli stack.

Associazione dei livelli di servizio Amazon RDS alle app

Dopo aver aggiunto un livello di servizio Amazon RDS, puoi associarlo a un'app.

- Puoi associare un layer Amazon RDS a un'app al momento della creazione dell'app o successivamente modificando la configurazione dell'app.
- Per dissociare un layer Amazon RDS da un'app, modifica la configurazione dell'app per specificare un server di database diverso o nessun server.

Il layer Amazon RDS rimane parte dello stack e può essere associato a un'app diversa.

Dopo aver associato un'istanza Amazon RDS a un'app, AWS OpsWorks Stacks inserisce le informazioni di connessione al database sui server dell'app. L'applicazione su ogni istanza del server può quindi utilizzare queste informazioni per collegarsi al database. Per ulteriori informazioni su come connettersi a un'istanza Amazon RDS, consultathe section called “Connessione a un database”.

Rimuovere un Amazon RDS Service Layer da uno stack

Per rimuovere un livello di servizio Amazon RDS da uno stack, devi annullarne la registrazione.

Per annullare la registrazione di un livello di servizio Amazon RDS

1. Fai clic su Livelli nel riquadro di navigazione e fai clic sul nome del livello di servizio Amazon RDS.
2. Fare clic su Deregister (Annulla registrazione) e confermare che si desidera annullare la registrazione del livello.

Questa procedura rimuove il layer dallo stack, ma non elimina l'istanza Amazon RDS sottostante. L'istanza e qualsiasi database rimangono nell'account e possono essere registrati con altri stack. È necessario utilizzare la console Amazon RDS, l'API o la CLI per eliminare l'istanza. Per ulteriori informazioni, consulta la sezione relativa all'eliminazione di un'istanza database.

Livelli del cluster ECS

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Il servizio Amazon Elastic Container Service (Amazon ECS) gestisce i container Docker in un cluster di istanze Amazon Elastic Compute Cloud (Amazon EC2), note come istanze container. Un livello

di cluster ECS rappresenta un cluster Amazon ECS e semplifica la gestione del cluster fornendo funzionalità che includono:

- Provisioning e gestione semplificati delle istanze di container
- Aggiornamenti di sistema operativo e pacchetti delle istanze di container
- Gestione delle autorizzazioni utente
- Monitoraggio delle prestazione delle istanze di container
- Amazon EBS (Amazon EBS) Gestione dei volumi Amazon EBS)
- Gestione degli indirizzi IP pubblici ed elastici
- Gestione dei gruppi di sicurezza

Il livello ECS Cluster ha le seguenti restrizioni e requisiti:

- Il livello è disponibile solo per gli stack di Chef 11.10 o Chef 12 Linux in esecuzione in un VPC, compreso un VPC predefinito.
- Le istanze del livello devono eseguire uno dei seguenti sistemi operativi.
  - Amazon Linux 2
  - Amazon Linux 2018.03
  - Amazon Linux 2017.09
  - Amazon Linux 2017.03
  - Amazon Linux 2016.09
  - Amazon Linux 2016.03
  - Amazon Linux 2015.09
  - Amazon Linux 2015.03
  - Ubuntu 18.04 LTS
  - Ubuntu 16.04 LTS
  - Ubuntu 14.04 LTS
  - Personalizza
- La versione dell'agente AWS OpsWorks Stacks nelle istanze del livello deve essere `3425-20150727112318` o successiva.

Argomenti

- Aggiungere un layer di cluster ECS a uno stack
- Gestione deli cluster ECS
- Eliminazione di un cluster ECS da uno stack

Aggiungere un layer di cluster ECS a uno stack

AWS OpsWorks Stacks semplifica il processo di avvio e manutenzione delle istanze di container per i cluster Amazon ECS esistenti. Per creare o avviare altre entità Amazon ECS, ad esempio cluster e task, utilizza la console Amazon ECS, l'interfaccia a riga di comando (CLI) o l'API Amazon ECS. (Per ulteriori informazioni, consulta la Guida per gli sviluppatori di Amazon Elastic Container Service.) Puoi quindi associare un cluster a uno stack creando un livello di cluster ECS, che puoi utilizzare per gestire il cluster in AWS OpsWorks Stacks.

Puoi associare i cluster agli stack come segue:

- Ogni stack può avere un livello ECS Cluster, che rappresenta un singolo cluster.
- Un cluster può essere associato solo a uno stack.

Prima di poter aggiungere layer ECS Cluster ai tuoi stack, devi aggiornare il ruolo del servizio AWS OpsWorks Stacks AWS Identity and Access Management (IAM), che di solito è denominato`aws-opsworks-service-role`, per consentire a AWS OpsWorks Stacks di interagire con Amazon ECS per tuo conto. Per ulteriori informazioni sul ruolo del servizio, consulta Consentire a AWS OpsWorks Stacks di agire per tuo conto.

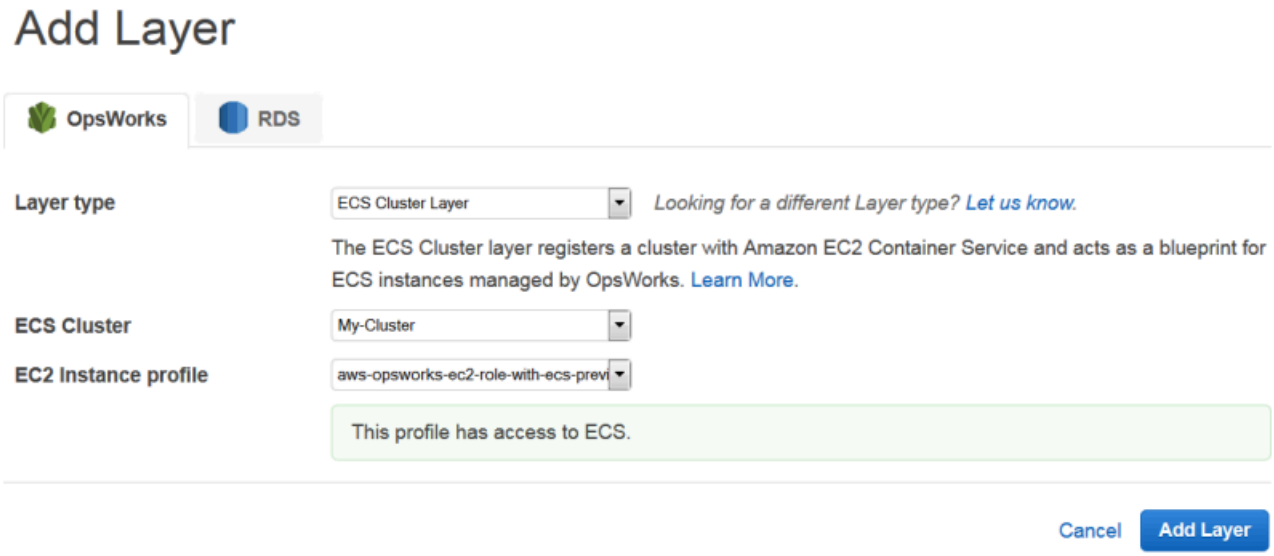
La prima volta che crei un layer ECS Cluster, la console fornisce un pulsante Update che puoi scegliere per indirizzare AWS OpsWorks Stacks ad aggiornare il ruolo al posto tuo. AWS OpsWorks Stacks visualizza quindi la pagina Aggiungi livello in modo da poter aggiungere il layer allo stack. Devi aggiornare il ruolo del servizio solo una volta. È quindi possibile utilizzare il ruolo aggiornato per aggiungere un layer ECS Cluster a qualsiasi stack.

> **Note**
>
> Se preferisci, puoi aggiornare manualmente la policy del ruolo del servizio aggiungendo l'autorizzazione `ecs:*` alla policy esistente, come segue:

```
{
  "Statement": [
```

```
    {
      "Action": [
        "ec2:*",
        "iam:PassRole",
        "cloudwatch:GetMetricStatistics",
        "elasticloadbalancing:*",
        "rds:*",
        "ecs:*"
      ],
      "Effect": "Allow",
      "Resource": ["*"]
    }
  ]
}
```

L'associazione di un cluster a uno stack richiede due operazioni, ovvero la registrazione del cluster con lo stack e quindi la creazione del livello associato. La console AWS OpsWorks Stacks combina questi passaggi; la creazione del livello registra automaticamente il cluster specificato. Se utilizzi l'API AWS OpsWorks Stacks, la CLI o l'SDK, devi utilizzare operazioni separate per registrare il cluster e creare il layer associato. Per utilizzare la console per aggiungere un layer ECS Cluster allo stack, scegli Livelli, scegli +Livello o Aggiungi un livello, quindi scegli il tipo di livello ECS Cluster.



La pagina Add Layer (Aggiungi livello) include le opzioni di configurazione seguenti:

Cluster ECS

Il cluster Amazon ECS che desideri registrare nello stack.

EC2 Profilo dell'istanza

Il profilo dell'istanza Amazon Elastic Compute Cloud (Amazon Amazon EC2) del cluster. Questo profilo concede l'autorizzazione alle applicazioni in esecuzione sulle istanze di container del cluster per accedere ad altri servizi AWS, incluso Amazon ECS. Quando crei il tuo primo layer ECS Cluster, scegli Nuovo profilo con accesso ECS a direct AWS OpsWorks Stacks per creare il profilo richiesto, denominato. `aws-opsworks-ec2-role-with-ecs` È quindi possibile utilizzare quel profilo per tutti i livelli ECS Cluster successivi. Per ulteriori informazioni sul profilo dell'istanza, consulta Specificazione delle autorizzazioni per le app in esecuzione su istanze EC2 .

Puoi specificare altre impostazioni modificando la configurazione del livello, ad esempio:

- Come collegare un sistema di bilanciamento del carico Elastic Load Balancing al livello.

  Questo approccio potrebbe essere adatto per alcuni casi d'uso, ma Amazon ECS offre opzioni più sofisticate. Per ulteriori informazioni, consulta Bilanciamento del carico nel servizio.
- Impostando l'assegnazione automatica di indirizzi IP pubblici o indirizzi IP elastici alle istanze di container.

  Se disabiliti l'assegnazione automatica per entrambi i tipi di indirizzi, l'istanza risulterà online solo se la sottorete dispone di un NAT configurato correttamente. Per ulteriori informazioni, consulta Esecuzione di uno stack in un VPC.

Gestione deli cluster ECS

Dopo aver creato un livello di cluster ECS, puoi utilizzare AWS OpsWorks Stacks per gestire il cluster nel modo seguente:

Provisioning e gestione delle istanze di container

Inizialmente, un livello di cluster ECS non include alcuna istanza di contenitore, anche se il cluster originale lo era. Un'opzione prevede la gestione delle istanze del livello utilizzando una combinazione delle seguenti operazioni:

- Aggiunta manuale di istanze 24 ore su 24, 7 giorni su 7 al livello e loro eliminazione quando non sono più necessarie.
- Aggiunta o eliminazione di istanze in base a una pianificazione mediante l'aggiunta di istanze basate sul tempo al livello.

- Aggiungi o elimina istanze in base alle metriche o agli CloudWatch allarmi AWS OpsWorks degli host di Stacks aggiungendo istanze basate sul carico al layer.

> **Note**
>
> Se Amazon ECS non è supportato per il sistema operativo predefinito dello stack, devi specificare esplicitamente un sistema operativo supportato: Amazon Linux 2, Amazon Linux 2018.03, Amazon Linux 2017.09, Amazon Linux 2017.03, Amazon Linux 2016.09, Amazon Linux 2016.03, Amazon Linux 2015.09, Amazon Linux 2015.03, Ubuntu 18.04 LTS, Ubuntu 16.04 LTS, Ubuntu 16.04 LTS, Ubuntu 14.04 LTS o Custom: quando si creano le istanze del contenitore. Non utilizzare l'AMI ottimizzata ECS per creare istanze in un livello ECS, poiché questa AMI include già l'agente ECS. AWS OpsWorks Stacks tenta inoltre di installare l'agente ECS durante il processo di configurazione dell'istanza e il conflitto può causare il fallimento della configurazione.

Per ulteriori informazioni, consultaOttimizzazione del numero di server. AWS OpsWorks Stacks assegna il gruppo di sicurezza AWS- OpsWorks -ECS-Cluster a ciascuna istanza. Al termine dell'avvio di ogni nuova istanza, AWS OpsWorks Stacks la converte in un'istanza container installando Docker e l'agente Amazon ECS, quindi registrando l'istanza nel cluster.

Se preferisci utilizzare istanze di container esistenti, puoi registrarle nello stack e assegnarle al livello ECS Cluster. Ricorda che le istanze devono eseguire un sistema operativo supportato, Amazon Linux 2015.03 o versioni successive o Ubuntu 14.04 LTS o versioni successive.

> **Note**
>
> Un'istanza di contenitore non può appartenere sia a un livello di cluster ECS che a un altro livello integrato. Tuttavia, un'istanza del contenitore può appartenere a un livello di cluster ECS e a uno o più livelli personalizzati.

Eseguire aggiornamenti del sistema operativo e dei pacchetti

Al termine dell'avvio di una nuova istanza, AWS OpsWorks Stacks installa gli aggiornamenti più recenti. Puoi quindi utilizzare AWS OpsWorks Stacks per mantenere aggiornate le istanze del contenitore. Per ulteriori informazioni, consulta Gestione degli aggiornamenti della sicurezza.

Gestione delle autorizzazioni utente

AWS OpsWorks Stacks offre un modo semplice per gestire le autorizzazioni sulle istanze del contenitore, inclusa la gestione delle chiavi SSH degli utenti. Per ulteriori informazioni, consultare Gestione delle autorizzazioni utente e Gestione dell'accesso SSH.

Monitoraggio dei parametri relativi alle prestazioni

AWS OpsWorks Stacks offre una varietà di modi per monitorare le metriche delle prestazioni per lo stack, il layer o le singole istanze. Per ulteriori informazioni, consulta Monitoraggio.

Gestisci altre attività di gestione, come la creazione di attività o servizi, tramite Amazon ECS. Per ulteriori informazioni, consulta la Guida per lo sviluppatore di Amazon Elastic Container.

> **Note**
>
> Per accedere direttamente alla pagina del cluster sulla console Amazon ECS, scegli Istanze, quindi scegli ECS Cluster, che si trova vicino all'angolo in alto a destra della sezione del livello ECS Cluster.

Eliminazione di un cluster ECS da uno stack

Quando il cluster non è più necessario, elimina il livello ECS Cluster e annulla la registrazione del cluster associato. La rimozione di un cluster da uno stack richiede due operazioni, ovvero l'annullamento della registrazione del cluster e quindi l'eliminazione del livello associato. La console AWS OpsWorks Stacks combina questi passaggi; l'eliminazione del livello annulla automaticamente la registrazione del cluster specificato. Se utilizzi l'API AWS OpsWorks Stacks, la CLI o l'SDK, devi utilizzare operazioni separate per annullare la registrazione del cluster ed eliminare il layer associato.

Per utilizzare la console per eliminare un layer ECS Cluster

1. Se desideri controllare la modalità di interruzione delle attività, utilizza la console, l'API o la CLI di Amazon ECS per ridimensionare ed eliminare i servizi del cluster. Per ulteriori informazioni, consulta Cleaning Up Your Amazon ECS Resources.
2. Arrestare le istanze del livello, quindi eliminarle. Quando interrompi un'istanza di container, AWS OpsWorks Stacks interrompe automaticamente tutte le attività in esecuzione, annulla la registrazione dell'istanza dal cluster e termina l'istanza.

> **Note**
>
> Se le istanze di container esistenti sono state registrate con lo stack, è possibile annullare l'assegnazione delle istanze al livello e quindi annullare la relativa registrazione. In questo modo le istanze tornano sotto il controllo di ECS.

3. Eliminare il layer. AWS OpsWorks Stacks annulla la registrazione del cluster associato, ma non lo elimina. Il cluster rimane in Amazon ECS.

AWS OpsWorks Stacks Layers personalizzati

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Un livello personalizzato ha solo una serie minima di ricette. È quindi possibile aggiungere funzionalità appropriate al livello implementando le ricette personalizzate e assegnandole agli eventi del ciclo di vita del livello.

Il livello personalizzato ha le seguenti impostazioni di configurazione.

> **Note**
>
> AWS OpsWorks Stacks installa automaticamente Ruby sulle istanze del layer. Se si desidera eseguire il codice Ruby sull'istanza, ma non si vuole utilizzare la versione predefinita di Ruby, è possibile utilizzare JSON personalizzato o un file di attributi personalizzato per specificare la propria versione preferita. Per ulteriori informazioni, consulta Versioni di Ruby.

La procedura di base per la creazione di un livello personalizzato ha i seguenti passaggi:

1. Implementazione di un libro di ricette che contiene le ricette e i file associati necessari per installare e configurare i pacchetti, gestire le modifiche di configurazione, distribuire le applicazioni e così via.

   A seconda dei requisiti, potrebbero anche essere necessarie ricette per gestire le attività di annullamento e di arresto. Per ulteriori informazioni, consulta Libri di ricette e ricette.
2. Creare un livello personalizzato.
3. Assegnare le proprie ricette agli eventi del ciclo di vita appropriati.

È quindi possibile aggiungere istanze al livello, avviarle e distribuire applicazioni a tali istanze.

> **Important**
>
> Per distribuire applicazioni alle istanze di un livello personalizzato, è necessario implementare le ricette per gestire l'operazione di distribuzione e assegnarle all'evento Distribuzione del livello.

Installazioni pacchetti del sistema operativo per livello

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

A partire da Chef 12, è necessario utilizzare ricette personalizzate per installare pacchetti sui livelli che eseguono sistemi operativi diversi. Questo approccio offre la massima flessibilità e il controllo sull'installazione di pacchetti.

Ad esempio, supponiamo di voler installare Apache su livelli che eseguono RedHat versioni Ubuntu e Amazon del sistema operativo Linux. Il pacchetto Apache per RedHat Amazon Linux si chiama`httpd`, ma su Ubuntu si chiama`apache2`.

Per risolvere la differenza nella denominazione dei pacchetti, è possibile utilizzare una sintassi simile a quella della seguente ricetta di esempio. La ricetta installa il pacchetto Apache appropriato per ogni sistema operativo. Questo esempio si basa sulla Documentazione Chef.

```
package "Install Apache" do
   case node[:platform]
      when "redhat", "amazon"
         package_name "httpd"
      when "ubuntu"
         package_name "apache2"
   end
end
```

Per informazioni dettagliate su come utilizzare la risorsa `package` per gestire i pacchetti, andare alla pagina package (pacchetto) nella documentazione Chef.

In alternativa, è possibile utilizzare il metodo helper `value_for_platform` da Chef Recipe DSL (linguaggio specifico di dominio), che esegue la stessa operazione in maniera più succinta:

```
package "Install Apache" do
   package_name value_for_platform(
      ["redhat", "amazon"] => { "default" => "httpd" },
      ["ubuntu"] => { "default" => "apache2" }
   )
end
```

Per informazioni su come utilizzare il metodo helper `value_for_platform` visitare Informazioni sulla ricetta DSL.

Istanze

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Un'istanza rappresenta una risorsa di elaborazione, ad esempio un' EC2 istanza Amazon, che gestisce il lavoro di fornitura delle applicazioni, il bilanciamento del traffico e così via. Il sistema operativo di un'istanza può avere una qualsiasi delle numerose distribuzioni Linux o Windows Server 2012 R2.

Le istanze possono essere aggiunte a uno stack in uno dei seguenti modi:

- Usa AWS OpsWorks Stacks per aggiungere istanze a uno stack. Le istanze che aggiungi rappresentano EC2 istanze Amazon.
- Per gli stack basati su Linux, puoi registrare istanze create altrove, incluse le istanze create con EC2 Amazon e le istanze locali in esecuzione sul tuo hardware.

  Puoi quindi utilizzare AWS OpsWorks Stacks per gestire queste istanze più o meno allo stesso modo delle istanze create con Stacks AWS OpsWorks

Questa sezione descrive come utilizzare AWS OpsWorks Stacks per creare e gestire istanze.

Argomenti

- Utilizzo delle AWS OpsWorks istanze Stacks
- Utilizzo di risorse informatiche create al di fuori di AWS OpsWorks Stacks
- Modifica della configurazione dell'istanza
- Eliminazione delle AWS OpsWorks istanze di Stacks
- Utilizzo di SSH per l'accesso a un'istanza Linux
- Utilizzo di RDP per l'accesso a un'istanza Windows

Utilizzo delle AWS OpsWorks istanze Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi usare AWS OpsWorks Stacks per creare istanze e aggiungerle allo stack.

Argomenti

- AWS OpsWorks Sistemi operativi impilati
- Aggiunta di un'istanza a un livello
- Utilizzo di Custom AMIs
- Avvio, arresto e riavvio manuali delle istanze 24 ore su 24, 7 giorni su 7
- Gestione del carico con istanze basate sul tempo e sul carico

AWS OpsWorks Sistemi operativi impilati

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks supporta le versioni a 64 bit di diversi sistemi operativi integrati, tra cui le distribuzioni Linux Amazon e Ubuntu e Microsoft Windows Server. Alcune note generali:

- Sulle istanze di uno stack è possibile eseguire Linux o Windows.

  Uno stack può includere versioni o distribuzioni Linux diverse su istanze differenti, ma non è possibile combinare le istanze Linux e Windows.
- È possibile utilizzare modelli personalizzati AMIs (Amazon Machine Images), ma devono essere basati su uno degli AWS OpsWorks stack AMIs supportati descritti negli argomenti di questa sezione. Sebbene sia possibile creare o registrare istanze con altri sistemi operativi (come CentOS 6). x) che sono stati creati in modo personalizzato o generato dalla community AMIs, non sono supportati ufficialmente.
  - Sistemi operativi Linux
  - Microsoft Windows Server
- Puoi avviare e arrestare le istanze manualmente o permettere a AWS OpsWorks Stacks di dimensionare automaticamente il numero delle istanze.

Puoi usare il dimensionamento automatico basato sul tempo con qualsiasi stack. Gli stack Linux, inoltre, sono in grado di eseguire il dimensionamento in base al carico.

- Oltre a utilizzare AWS OpsWorks Stacks per creare istanze Amazon, puoi anche registrare EC2 istanze con uno stack Linux che sono state create al di fuori di Stacks. AWS OpsWorks

  Ciò include EC2 le istanze Amazon e le istanze eseguite sul tuo hardware. Tuttavia, devono eseguire una delle distribuzioni Linux supportate. Non puoi registrare istanze Amazon EC2 o Windows locali.

Puoi eseguire l'`DescribeOperatingSystems`API AWS OpsWorks Stacks per restituire un elenco di sistemi operativi supportati e le relative versioni supportate di Chef. Di seguito è riportato un comando di esempio con AWS CLI.

```
aws opsworks describe-operating-systems
```

Di seguito è riportata una risposta di esempio.

```
{
    "OperatingSystems": [
        {
            "Name": "Amazon Linux",
            "Id": "Amazon Linux",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "11.10"
                },
                {
                    "Name": "Chef",
                    "Version": "11.4"
                },
                {
                    "Name": "Chef",
                    "Version": "0.9"
                }
            ],
            "ReportedName": "amazon",
            "ReportedVersion": "2014.03",
            "Supported": false
```

```
        },
        {
            "Name": "Amazon Linux 2",
            "Id": "Amazon Linux 2",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                }
            ],
            "ReportedName": "amazon",
            "ReportedVersion": "2"
        },
        {
            "Name": "Amazon Linux 2014.09",
            "Id": "Amazon Linux 2014.09",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "11.10"
                },
                {
                    "Name": "Chef",
                    "Version": "11.4"
                },
                {
                    "Name": "Chef",
                    "Version": "0.9"
                }
            ],
            "ReportedName": "amazon",
            "ReportedVersion": "2014.09",
            "Supported": false
        },
        {
            "Name": "Amazon Linux 2015.03",
            "Id": "Amazon Linux 2015.03",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
```

```
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                },
                {
                    "Name": "Chef",
                    "Version": "11.4"
                },
                {
                    "Name": "Chef",
                    "Version": "0.9"
                }
            ],
            "ReportedName": "amazon",
            "ReportedVersion": "2015.03",
            "Supported": false
        },
        {
            "Name": "Amazon Linux 2015.09",
            "Id": "Amazon Linux 2015.09",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                },
                {
                    "Name": "Chef",
                    "Version": "11.4"
                },
                {
                    "Name": "Chef",
                    "Version": "0.9"
                }
            ],
            "ReportedName": "amazon",
            "ReportedVersion": "2015.09",
            "Supported": false
        },
```

```
        {
            "Name": "Amazon Linux 2016.03",
            "Id": "Amazon Linux 2016.03",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                },
                {
                    "Name": "Chef",
                    "Version": "11.4"
                },
                {
                    "Name": "Chef",
                    "Version": "0.9"
                }
            ],
            "ReportedName": "amazon",
            "ReportedVersion": "2016.03"
        },
        {
            "Name": "Amazon Linux 2016.09",
            "Id": "Amazon Linux 2016.09",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                },
                {
                    "Name": "Chef",
                    "Version": "11.4"
                },
                {
                    "Name": "Chef",
```

```
                    "Version": "0.9"
                }
            ],
            "ReportedName": "amazon",
            "ReportedVersion": "2016.09"
        },
        {
            "Name": "Amazon Linux 2017.03",
            "Id": "Amazon Linux 2017.03",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                },
                {
                    "Name": "Chef",
                    "Version": "11.4"
                },
                {
                    "Name": "Chef",
                    "Version": "0.9"
                }
            ],
            "ReportedName": "amazon",
            "ReportedVersion": "2017.03"
        },
        {
            "Name": "Amazon Linux 2017.09",
            "Id": "Amazon Linux 2017.09",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                },
```

```
                {
                    "Name": "Chef",
                    "Version": "11.4"
                },
                {
                    "Name": "Chef",
                    "Version": "0.9"
                }
            ],
            "ReportedName": "amazon",
            "ReportedVersion": "2017.09"
        },
        {
            "Name": "Amazon Linux 2018.03",
            "Id": "Amazon Linux 2018.03",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                }
            ],
            "ReportedName": "amazon",
            "ReportedVersion": "2018.03"
        },
        {
            "Name": "CentOS Linux 7",
            "Id": "CentOS Linux 7",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                }
            ],
            "ReportedName": "CentOS Linux",
            "ReportedVersion": "7"
        },
        {
            "Name": "Microsoft Windows Server 2012 R2 Base",
```

```
            "Id": "Microsoft Windows Server 2012 R2 Base",
            "Type": "Windows",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12.2"
                }
            ],
            "ReportedName": "microsoft windows server",
            "ReportedVersion": "2012 r2 standard",
            "Supported": false
        },
        {
            "Name": "Microsoft Windows Server 2012 R2 with SQL Server Express",
            "Id": "Microsoft Windows Server 2012 R2 with SQL Server Express",
            "Type": "Windows",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12.2"
                }
            ],
            "ReportedName": "microsoft windows server",
            "ReportedVersion": "2012 r2 standard",
            "Supported": false
        },
        {
            "Name": "Microsoft Windows Server 2012 R2 with SQL Server Standard",
            "Id": "Microsoft Windows Server 2012 R2 with SQL Server Standard",
            "Type": "Windows",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12.2"
                }
            ],
            "ReportedName": "microsoft windows server",
            "ReportedVersion": "2012 r2 standard",
            "Supported": false
        },
        {
            "Name": "Microsoft Windows Server 2012 R2 with SQL Server Web",
            "Id": "Microsoft Windows Server 2012 R2 with SQL Server Web",
            "Type": "Windows",
```

```
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12.2"
                }
            ],
            "ReportedName": "microsoft windows server",
            "ReportedVersion": "2012 r2 standard",
            "Supported": false
        },
        {
            "Name": "Microsoft Windows Server 2019 Base",
            "Id": "Microsoft Windows Server 2019 Base",
            "Type": "Windows",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12.2"
                }
            ],
            "ReportedName": "microsoft windows server",
            "ReportedVersion": "2019 datacenter"
        },
        {
            "Name": "Microsoft Windows Server 2019 with SQL Server Express",
            "Id": "Microsoft Windows Server 2019 with SQL Server Express",
            "Type": "Windows",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12.2"
                }
            ],
            "ReportedName": "microsoft windows server",
            "ReportedVersion": "2019 datacenter"
        },
        {
            "Name": "Microsoft Windows Server 2019 with SQL Server Standard",
            "Id": "Microsoft Windows Server 2019 with SQL Server Standard",
            "Type": "Windows",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12.2"
```

```
            }
        ],
        "ReportedName": "microsoft windows server",
        "ReportedVersion": "2019 datacenter"
    },
    {
        "Name": "Microsoft Windows Server 2019 with SQL Server Web",
        "Id": "Microsoft Windows Server 2019 with SQL Server Web",
        "Type": "Windows",
        "ConfigurationManagers": [
            {
                "Name": "Chef",
                "Version": "12.2"
            }
        ],
        "ReportedName": "microsoft windows server",
        "ReportedVersion": "2019 datacenter"
    },
    {
        "Name": "Microsoft Windows Server 2022 Base",
        "Id": "Microsoft Windows Server 2022 Base",
        "Type": "Windows",
        "ConfigurationManagers": [
            {
                "Name": "Chef",
                "Version": "12.2"
            }
        ],
        "ReportedName": "microsoft windows server",
        "ReportedVersion": "2022 datacenter"
    },
    {
        "Name": "Microsoft Windows Server 2022 with SQL Server Express",
        "Id": "Microsoft Windows Server 2022 with SQL Server Express",
        "Type": "Windows",
        "ConfigurationManagers": [
            {
                "Name": "Chef",
                "Version": "12.2"
            }
        ],
        "ReportedName": "microsoft windows server",
        "ReportedVersion": "2022 datacenter"
    },
```

```
        {
            "Name": "Microsoft Windows Server 2022 with SQL Server Standard",
            "Id": "Microsoft Windows Server 2022 with SQL Server Standard",
            "Type": "Windows",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12.2"
                }
            ],
            "ReportedName": "microsoft windows server",
            "ReportedVersion": "2022 datacenter"
        },
        {
            "Name": "Microsoft Windows Server 2022 with SQL Server Web",
            "Id": "Microsoft Windows Server 2022 with SQL Server Web",
            "Type": "Windows",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12.2"
                }
            ],
            "ReportedName": "microsoft windows server",
            "ReportedVersion": "2022 datacenter"
        },
        {
            "Name": "Red Hat Enterprise Linux 7",
            "Id": "Red Hat Enterprise Linux 7",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                }
            ],
            "ReportedName": "Red Hat Enterprise Linux",
            "ReportedVersion": "7"
        },
        {
```

```
            "Name": "Ubuntu 12.04 LTS",
            "Id": "Ubuntu 12.04 LTS",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                },
                {
                    "Name": "Chef",
                    "Version": "11.4"
                },
                {
                    "Name": "Chef",
                    "Version": "0.9"
                }
            ],
            "ReportedName": "ubuntu",
            "ReportedVersion": "12.04",
            "Supported": false
        },
        {
            "Name": "Ubuntu 14.04 LTS",
            "Id": "Ubuntu 14.04 LTS",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                }
            ],
            "ReportedName": "ubuntu",
            "ReportedVersion": "14.04"
        },
        {
            "Name": "Ubuntu 16.04 LTS",
```

```
            "Id": "Ubuntu 16.04 LTS",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                }
            ],
            "ReportedName": "ubuntu",
            "ReportedVersion": "16.04"
        },
        {
            "Name": "Ubuntu 18.04 LTS",
            "Id": "Ubuntu 18.04 LTS",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                }
            ],
            "ReportedName": "ubuntu",
            "ReportedVersion": "18.04"
        },
        {
            "Name": "Ubuntu 20.04 LTS",
            "Id": "Ubuntu 20.04 LTS",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12"
                }
            ],
            "ReportedName": "ubuntu",
            "ReportedVersion": "20.04"
        },
        {
            "Name": "Custom",
            "Id": "Custom",
            "Type": "Linux",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
```

```
                    "Version": "12"
                },
                {
                    "Name": "Chef",
                    "Version": "11.10"
                },
                {
                    "Name": "Chef",
                    "Version": "11.4"
                },
                {
                    "Name": "Chef",
                    "Version": "0.9"
                }
            ]
        },
        {
            "Name": "CustomWindows",
            "Id": "CustomWindows",
            "Type": "Windows",
            "ConfigurationManagers": [
                {
                    "Name": "Chef",
                    "Version": "12.2"
                }
            ]
        }
    ]
}
```

Argomenti

- Sistemi operativi Linux
- Microsoft Windows Server

Sistemi operativi Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks supporta le versioni a 64 bit dei seguenti sistemi operativi Linux.

- Amazon Linux e Amazon Linux 2 (consulta la console AWS OpsWorks Stacks per le versioni attualmente supportate)
- Ubuntu 20.04 LTS
- CentOS 7
- Red Hat Enterprise Linux 7

È inoltre possibile utilizzare opzioni personalizzate AMIs in base a questi sistemi operativi.

Alcune note generali sulle istanze Linux:

Versioni dei pacchetti supportate

Le versioni supportate e i livelli di patch per i pacchetti, come Ruby, dipendono dal sistema operativo e dalla versione, come descritto nelle seguenti sezioni.

Aggiornamenti

Per impostazione predefinita, AWS OpsWorks Stacks garantisce che le istanze Linux dispongano delle patch di sicurezza più recenti chiamando automaticamente `yum update` o `apt-get update` dopo l'avvio di un'istanza. Per disabilitare gli aggiornamenti automatici CreateInstance, utilizza UpdateLayerle azioni UpdateInstanceCreateLayer,, o i metodi SDK AWS equivalenti o i comandi AWS CLI su cui impostare il parametro. `InstallUpdatesOnBoot` `false`

Per evitare interruzioni del servizio, AWS OpsWorks Stacks non installa automaticamente gli aggiornamenti dopo che un'istanza è online. Puoi aggiornare manualmente il sistema operativo di un'istanza online in qualsiasi momento, eseguendo il comando dello stack Upgrade Operating System (Aggiorna sistema operativo). Per ulteriori informazioni su come gestire gli aggiornamenti di sicurezza, consulta Gestione degli aggiornamenti della sicurezza.

Per un maggiore controllo sul modo in cui AWS OpsWorks Stacks aggiorna le tue istanze, crea un'AMI personalizzata basata su uno dei sistemi operativi supportati. Ad esempio, con custom AMIs puoi specificare quali versioni del pacchetto sono installate su un'istanza. Ogni distribuzione Linux dispone di diversi tempi di assistenza e policy di unione dei pacchetti, pertanto devi valutare

l'approccio che meglio soddisfa i tuoi requisiti. Per ulteriori informazioni, consulta Utilizzo di Custom AMIs.

File Hosts

Ogni istanza online ha un `/etc/hosts` file che associa gli indirizzi IP ai nomi degli host. AWS OpsWorks Stacks include gli indirizzi pubblici e privati per tutte le istanze online dello stack nel file di ogni istanza. `hosts` Ad esempio, supponiamo di avere uno stack con due istanze di Node.js App Server, nodejs-app1 e nodejs-app2, e un'istanza MySQL, db-master1. Il file `hosts` dell'istanza nodejs-app1 sarà simile all'esempio seguente e quelli delle altre istanze avranno file `hosts` simili.

```
...
# OpsWorks Layer State
192.0.2.0 nodejs-app1.localdomain nodejs-app1
10.145.160.232 db-master1
198.51.100.0 db-master1-ext
10.243.77.78 nodejs-app2
203.0.113.0 nodejs-app2-ext
10.84.66.6 nodejs-app1
192.0.2.0 nodejs-app1-ext
```

AWS OpsWorks Supporto proxy per agenti Stacks

L'agente AWS OpsWorks Stacks per Chef 11.10 e stack successivi include il supporto di base per i server proxy, che vengono generalmente utilizzati con sistemi isolati. VPCs Per abilitare il supporto del server proxy, un'istanza deve disporre di un file `/etc/environment` che fornisce le impostazioni appropriate per il traffico HTTP e HTTPS. Il file dovrebbe essere simile al seguente, sapendo che puoi sostituire il testo evidenziato con la porta e l'URL del server proxy:

```
http_proxy="http://myproxy.example.com:8080/"
https_proxy="http://myproxy.example.com:8080/"
no_proxy="169.254.169.254"
```

Per abilitare il supporto proxy, ti consigliamo di creare un'AMI personalizzata che includa un file `/etc/environment` appropriato e di utilizzare tale AMI per creare le tue istanze.

> **Note**
>
> Non è consigliabile utilizzare una ricetta personalizzata per creare un `/etc/environment` file sulle istanze. AWS OpsWorks Stacks necessita dei dati del server proxy nelle prime fasi del processo di configurazione, prima dell'esecuzione di qualsiasi ricetta personalizzata.

Argomenti

- Amazon Linux
- Ubuntu LTS
- CentOS
- Red Hat Enterprise Linux

Amazon Linux

AWS OpsWorks Stacks supporta le versioni a 64 bit di Amazon Linux e Amazon Linux 2. Oltre ad aggiornamenti e patch regolari, Amazon Linux rilascia una nuova versione all'incirca ogni sei mesi. Questo può comportare modifiche rilevanti. Quando crei uno stack o una nuova istanza, devi specificare la versione di Amazon Linux da utilizzare. Quando AWS rilascia una nuova versione, le tue istanze continuano a eseguire la versione specificata finché non la modificherai esplicitamente. Dopo il rilascio di una nuova versione di Amazon Linux, vi sarà un periodo di migrazione di quattro settimane, durante il quale AWS continua a fornire aggiornamenti regolari per la versione precedente. Al termine del periodo di migrazione, le istanze possono continuare a eseguire la versione precedente, ma AWS non fornirà ulteriori aggiornamenti. Per ulteriori informazioni, consulta AMI Amazon Linux FAQs.

Quando viene rilasciata una nuova versione di Amazon Linux, ti consigliamo di eseguire l'aggiornamento alla nuova versione entro il periodo di migrazione, in modo che le istanze continuino a ricevere gli aggiornamenti della sicurezza. Prima di aggiornare le istanze del tuo stack di produzione, ti consigliamo di avviare una nuova istanza e verificare che la tua app venga eseguita correttamente sulla nuova versione. Dopodiché potrai aggiornare le istanze dello stack di produzione.

> **Note**
>
> Per impostazione predefinita, le versioni personalizzate AMIs basate su Amazon Linux vengono aggiornate automaticamente alla nuova versione quando viene rilasciata. La pratica

consigliata consiste nel bloccare l'AMI personalizzata a una specifica versione di Amazon Linux in modo da rimandare l'aggiornamento finché non avrai testato la nuova versione. Per ulteriori informazioni, consulta Come blocco l'AMI a una versione specifica?
Se utilizzi un AWS CloudFormation modello per creare stack con istanze che eseguono Amazon Linux, i modelli devono specificare esplicitamente una versione di Amazon Linux. In particolare, se il modello specifica `Amazon Linux`, le istanze continueranno a eseguire la versione 2016.09. Per ulteriori informazioni, consulta AWS::OpsWorks::Stack e AWS::OpsWorks::Instance.

Per aggiornare la versione di Amazon Linux di un'istanza, procedi in uno dei seguenti modi:

- Per le istanze online, esegui il comando dello stack Upgrade Operating System (Aggiorna sistema operativo).

  Quando è disponibile una nuova versione di Amazon Linux, le pagine Instances (Istanze) e Stack mostrano un avviso con un collegamento che consente di visualizzare la pagina Run Command (Esegui comando). Quindi, puoi eseguire Upgrade Operating System (Aggiorna sistema operativo) per effettuare l'aggiornamento della tua istanza.
- Per le istanze offline supportate da Amazon Elastic Block Store (EBS), avvia le istanze ed esegui Upgrade Operating System, come descritto nell'istruzione precedente.
- Nel caso di istanze offline supportate da instance store, incluse le istanze basate sul tempo e sul carico, modifica l'impostazione Operating system (Sistema operativo) dell'istanza in modo che venga specificata la nuova versione.

  AWS OpsWorks Stacks aggiorna automaticamente le istanze alla nuova versione quando vengono riavviate.

Amazon Linux: versioni di Node.js supportate

Versione di Amazon Linux	Versioni di Node.js
`2`	`(Not applicable to operating systems that are available for Chef 12 and higher stacks only)`

Versione di Amazon Linux	Versioni di Node.js
2018.03	0.12.18
2017.09	0.12.18
2017.03	0.12.18
2016.09	0.12.18 0.12.17 0.12.16 0.12.15
2016.03	0.12.18 0.12.17 0.12.16 0.12.15 0.12.14 0.12.13 0.12.12 0.12.10

Amazon Linux: versioni di Chef supportate

Versione di Chef	Versioni di Amazon Linux supportate
12	Amazon Linux 2 Amazon Linux 2018.03 Amazon Linux 2017.09 Amazon Linux 2017.03 Amazon Linux 2016.09 Amazon Linux 2016.03
11.10	Amazon Linux 2018.03 Amazon Linux 2017.09 Amazon Linux 2017.03

Versione di Chef	Versioni di Amazon Linux supportate
	Amazon Linux 2016.09 Amazon Linux 2016.03
11.4 (deprecated)	Amazon Linux 2016.09 Amazon Linux 2016.03

Important

Prima di aggiornare le istanze t1.micro, accertati di disporre di un file di scambio temporaneo, `/var/swapfile`. Le istanze t1.micro sugli stack Chef 0.9 non dispongono di un file di scambio. Per gli stack Chef 11.4 e Chef 11.10, le versioni recenti dell'agente dell'istanza creano automaticamente un file di scambio per le istanze t1.micro. Tuttavia, questa modifica è stata introdotta nel corso di diverse settimane, pertanto dovrai verificare l'esistenza di `/var/swapfile` sulle istanze create approssimativamente prima del 24 marzo 2014.
Se le istanze t1.micro non dispongono di un file di scambio, puoi crearne uno nel modo seguente:

- Per gli stack Chef 11.10 e versioni successive, crea nuove istanze t1.micro, che dispongono automaticamente di un file di scambio.
- Per gli stack Chef 0.9, esegui i comandi seguenti su ogni istanza come utente root.

```
dd if=/dev/zero of=/var/swapfile bs=1M count=256
 mkswap /var/swapfile
 chown root:root /var/swapfile
 chmod 0600 /var/swapfile
 swapon /var/swapfile
```

Puoi utilizzare questi comandi anche sugli stack Chef 11.10 e versioni successive se non desideri creare nuove istanze.

Ubuntu LTS

Ubuntu rilascia una nuova versione di Ubuntu LTS all'incirca ogni due anni e supporta ogni rilascio per circa cinque anni. Ubuntu fornisce patch di protezione e aggiornamenti per tutta la durata del supporto del sistema operativo. Per ulteriori informazioni, consulta LTS - Ubuntu Wiki.

- Non è possibile aggiornare un'istanza Ubuntu esistente a una versione più recente di Ubuntu.

  È necessario creare una nuova istanza di Ubuntu ed eliminare l'istanza precedente.
- Ubuntu 20.04 LTS è supportato solo per Chef 12 e stack successivi.

CentOS

AWS OpsWorks Stacks supporta la versione a 64 bit di CentOS 7. La versione supportata iniziale è CentOS 7. CentOS rilascia una nuova versione all'incirca ogni due anni.

Quando si avvia una nuova istanza in uno stack CentOS, AWS OpsWorks Stacks installa automaticamente la versione CentOS più recente. Poiché AWS OpsWorks Stacks non aggiorna automaticamente il sistema operativo sulle istanze esistenti quando viene rilasciata una nuova versione secondaria di CentOS, un'istanza appena creata potrebbe ricevere una versione più recente rispetto alle istanze esistenti dello stack. Per far sì che le versioni sullo stack siano coerenti, puoi aggiornare le istanze esistenti alla versione di CentOS attuale, come segue:

- Nel caso delle istanze online, esegui il comando dello stack Upgrade Operating System (Aggiorna sistema operativo), che esegue `yum update` sulle istanze specificate per aggiornarle alla versione corrente.

  Quando è disponibile una nuova versione secondaria di CentOS 7, le pagine Instances (Istanze) e Stack mostrano un avviso con un collegamento che consente di visualizzare la pagina Run Command (Esegui comando). Quindi, puoi eseguire Upgrade Operating System (Aggiorna sistema operativo) per effettuare l'aggiornamento delle tue istanze.
- Per le istanze offline supportate da Amazon EBS, avvia le istanze ed esegui Upgrade Operating System come descritto nella voce precedente dell'elenco.
- Per le istanze offline archiviate, AWS OpsWorks Stacks installa automaticamente la nuova versione al riavvio delle istanze.

CentOS: versioni di Chef supportate

Versione di Chef	Versione di CentOS supportata
`12`	`CentOS 7`
`11.10`	`(None supported)`
`11.4 (deprecated)`	`(None supported)`

> **Note**
>
> AWS OpsWorks Stacks supporta Apache 2.4 per istanze CentOS.

Red Hat Enterprise Linux

AWS OpsWorks Stacks supporta la versione a 64 bit di Red Hat Enterprise Linux 7 (RHEL 7). La versione supportata iniziale è RHEL 7.1. Red Hat rilascia una nuova versione secondaria all'incirca ogni 9 mesi. Le versioni secondarie devono essere compatibili con RHEL 7.0. Per ulteriori informazioni, visita la pagina dedicata a ciclo di vita e policy di aggiornamento.

Quando si avvia una nuova istanza, AWS OpsWorks Stacks installa automaticamente la versione corrente di RHEL 7. Poiché AWS OpsWorks Stacks non aggiorna automaticamente il sistema operativo sulle istanze esistenti quando viene rilasciata una nuova versione secondaria di RHEL 7, un'istanza appena creata potrebbe ricevere una versione più recente rispetto alle istanze esistenti dello stack. Per mantenere le versioni coerenti sullo stack, puoi aggiornare le tue istanze esistenti alla versione di RHEL 7 attuale, come segue:

- Nel caso delle istanze online, esegui il comando dello stack Upgrade Operating System (Aggiorna sistema operativo), che esegue `yum update` sulle istanze specificate per aggiornarle alla versione corrente.

  Quando è disponibile una nuova versione di RHEL 7, le pagine Instances (Istanze) e Stack mostrano un avviso con un collegamento che consente di visualizzare la pagina Run Command (Esegui comando). Quindi, puoi eseguire Upgrade Operating System (Aggiorna sistema operativo) per effettuare l'aggiornamento delle tue istanze.

- Per le istanze offline supportate da Amazon EBS, avvia le istanze ed esegui Upgrade Operating System come descritto nella voce precedente dell'elenco.
- Per le istanze offline archiviate, AWS OpsWorks Stacks installa automaticamente la nuova versione al riavvio delle istanze.

Red Hat Enterprise Linux: versioni supportate di Node.js

Versione RHEL	Versioni di Node.js
7	(Node.js versions only apply to Chef 11.10 stacks) 0.8.19 0.8.26 0.10.11 0.10.21 0.10.24 0.10.25 0.10.27 0.10.29 0.10.40 0.12.10 0.12.12 0.12.13 0.12.15

Red Hat Enterprise Linux: versioni di Chef supportate

Versione di Chef	Versione di RHEL supportata
12	Red Hat Enterprise Linux 7
11.10	Red Hat Enterprise Linux 7
11.4 (deprecated)	(None supported)

Tutte le versioni di Node.js precedenti alla 0.10.40 sono obsolete. Anche le versioni 0.12.7 e 0.12.9 sono obsolete.

> **Note**
>
> AWS OpsWorks Stacks supporta Apache 2.4 per istanze RHEL 7.

Microsoft Windows Server

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le note seguenti descrivono il supporto di AWS OpsWorks Stacks per le istanze Windows. Le istanze Windows sono disponibili solo per gli stack Chef 12.2. L'esatta versione di Chef in uno stack Windows è la 12.22.

Attualmente, l'agente AWS OpsWorks Stacks non può essere installato e AWS OpsWorks Stacks non può gestire istanze basate su Windows che utilizzano una lingua dell'interfaccia utente di sistema diversa dall'inglese - Stati Uniti d'America (en-US).

Versioni

AWS OpsWorks Stacks supporta le seguenti versioni di Windows a 64 bit:

- Microsoft Windows Server 2022 Base
- Microsoft Windows Server 2022 con SQL Server Express
- Microsoft Windows Server 2022 con SQL Server Standard
- Microsoft Windows Server 2022 con SQL Server Web
- Microsoft Windows Server 2019 Base
- Microsoft Windows Server 2019 con SQL Server Express

- Microsoft Windows Server 2019 con SQL Server Standard
- Microsoft Windows Server 2019 con SQL Server Web

Creazione di istanze

Puoi creare istanze Windows con la console AWS OpsWorks Stacks, l'API o la CLI. Le istanze Windows sono supportate da Amazon EBS, ma non è possibile montare volumi Amazon EBS aggiuntivi.

Gli stack Windows possono utilizzare istanze 24/7, che puoi avviare e arrestare manualmente. È inoltre possibile utilizzare la scalabilità automatica basata sul tempo, che avvia e arresta automaticamente le istanze in base a una pianificazione definita dall'utente. Gli stack basati su Windows non possono utilizzare la scalabilità automatica basata sul carico.

Non è possibile registrare istanze Windows create al di fuori di Stacks con uno AWS OpsWorks stack.

Aggiornamenti

AWS aggiorna Windows AMIs per ogni set di patch, quindi quando crei un'istanza, avrà gli aggiornamenti più recenti. Tuttavia, AWS OpsWorks Stacks non fornisce un modo per applicare gli aggiornamenti alle istanze Windows online. Il modo più semplice per assicurare che Windows sia aggiornato è sostituire regolarmente le istanze, affinché eseguano sempre l'AMI più recente.

Livelli

Per gestire attività quali l'installazione e la configurazione di software o la distribuzione di app, dovrai implementare uno o più livelli personalizzati con ricette personalizzate.

Chef

Le istanze Windows utilizzano Chef 12.22 ed eseguono chef-client nella modalità locale, che consente di avviare un server Chef in-memory locale chiamato chef-zero. La presenza di questo server permette alle ricette personalizzate di utilizzare i contenitori di dati e la ricerca di Chef.

Login remoto

AWS OpsWorks Stacks fornisce agli utenti IAM autorizzati una password che possono utilizzare per accedere alle istanze Windows. Questa password scade dopo un determinato periodo di tempo. Gli amministratori possono utilizzare una coppia di chiavi SSH per recuperare la password Amministratore di un'istanza, che offre un numero illimitato di accessi RDP. Per ulteriori informazioni, consulta Accesso con RDP.

SDK AWS

AWS OpsWorks Stacks installa automaticamente su ogni istanza. AWS SDK per .NET Questo pacchetto include le librerie AWS .NET e gli strumenti AWS per Windows, inclusi gli strumenti AWS per PowerShell. Per usare il kit SDK Ruby, puoi far sì che una ricetta personalizzata installi il pacchetto gem appropriato.

Monitoraggio e metrica

Le istanze Windows supportano i parametri standard di Amazon CloudWatch (CloudWatch), che puoi visualizzare nella CloudWatch console.

Ruby

Il client Chef 12.22 che AWS OpsWorks Stacks installa sulle istanze Windows viene fornito con Ruby 2.3.6. Tuttavia, AWS OpsWorks Stacks non aggiunge la directory dell'eseguibile alla variabile di ambiente PATH. Per far sì che le tue applicazioni utilizzino questa versione di Ruby, puoi generalmente reperirla in `C:\opscode\chef\embedded\bin\`.

AWS OpsWorks CLI di Stacks Agent

L'agente AWS OpsWorks Stacks sulle istanze Windows non espone un'interfaccia a riga di comando.

Supporto Proxy

Per configurare il supporto proxy per le istanze Windows, attieniti alla procedura seguente:

1. Modifica `machine.config` per aggiungere quanto segue, che aggiunge il supporto proxy alle applicazioni Windows PowerShell (bootstrap iniziale) e.NET (agente Stacks):AWS OpsWorks

```
<system.net>
  <defaultProxy>
    <proxy autoDetect="false" bypassonlocal="true"
 proxyaddress="http://10.100.1.91:3128"  usesystemdefault="false" />
    <bypasslist>
      <add address="localhost" />
      <add address="169.254.169.254" />
    </bypasslist>
  </defaultProxy>
</system.net>
```

2. Eseguire i seguenti comandi per impostare le variabili di ambiente che verranno utilizzate in seguito da Chef e Git:

```
setx /m no_proxy "localhost,169.254.169.254"
setx /m http_proxy "http://10.100.1.91:3128"
setx /m https_proxy "http://10.100.1.91:3128"
```

> **Note**
>
> Per un maggiore controllo sul modo in cui AWS OpsWorks Stacks aggiorna le tue istanze, crea un'AMI personalizzata basata su Microsoft Windows Server 2022 Base. Ad esempio, con custom AMIs puoi specificare quale software è installato su un'istanza, ad esempio Web Server (IIS). Per ulteriori informazioni, consulta Utilizzo di Custom AMIs.

Aggiunta di un'istanza a un livello

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver creato un livello, è comune aggiungere almeno un'istanza. Se l'insieme corrente non è in grado di gestire il carico, è possibile aggiungere altre istanze in un secondo momento. Puoi utilizzare anche le istanze basate su tempo o carico per ricalibrare automaticamente il numero di istanze.

Puoi aggiungere istanze nuove o esistenti a un livello:

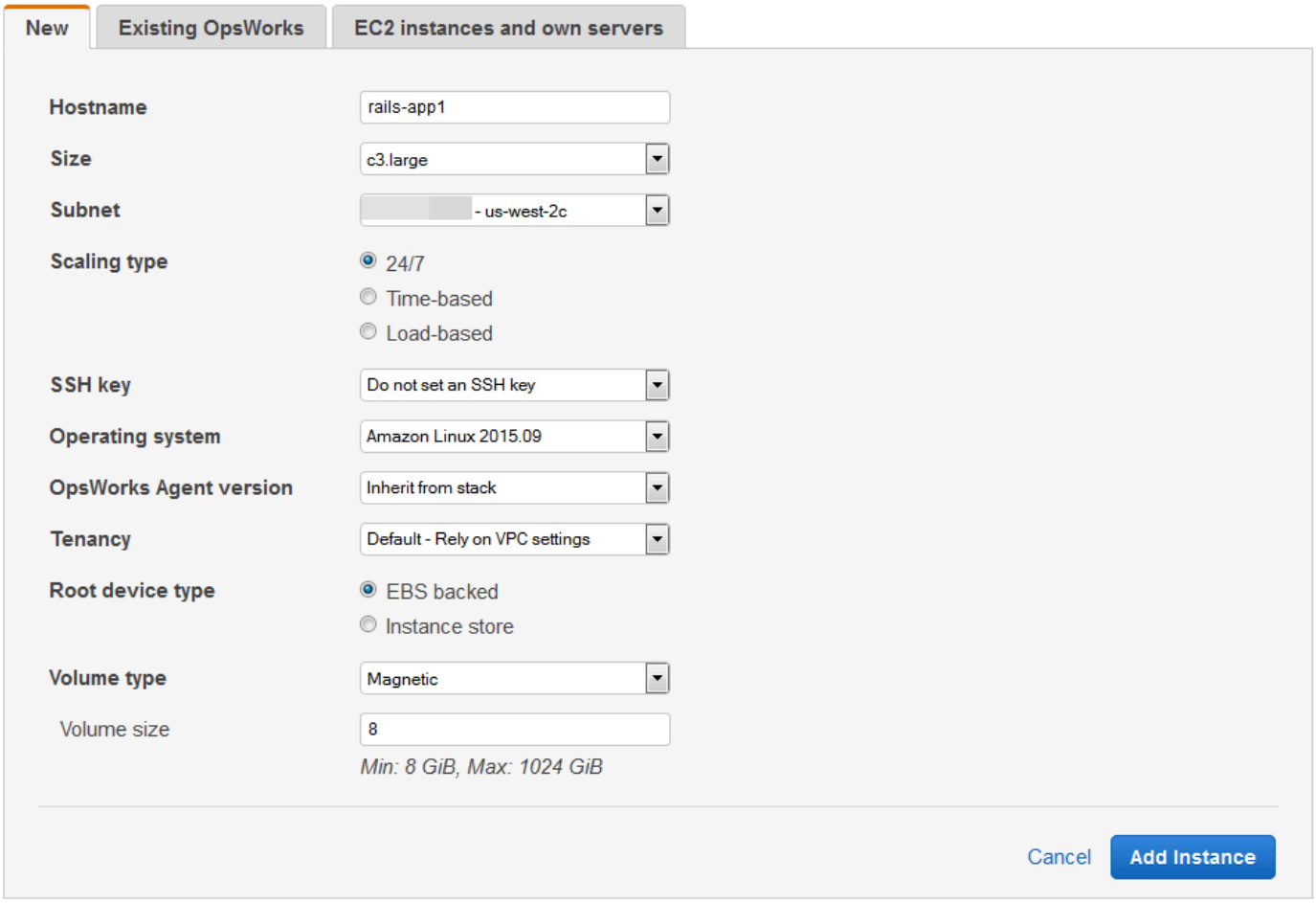
- Nuova: OpsWorks crea una nuova istanza, configurata secondo le tue specifiche e la rende un membro del layer.
- Esistente: è possibile aggiungere un'istanza esistente da qualsiasi livello compatibile, ma deve trovarsi nello stato offline (interrotto).

Se un'istanza appartiene a più livelli, AWS OpsWorks Stacks esegue le ricette per ciascuno dei livelli dell'istanza quando si verifica un evento del ciclo di vita oppure all'esecuzione di un comando relativo a stack o distribuzione.

Puoi anche rendere un'istanza membro di vari livelli modificandone la configurazione. Per ulteriori informazioni, consulta Modifica della configurazione dell'istanza.

Per aggiungere una nuova istanza a un livello

1. Nella pagina Instances (Istanze), scegliere +Instance (+Istanza) per il livello appropriato e (se necessario) scegliere la scheda New (Nuovo). Per configurare altri parametri oltre a Host name (Nome host), Size (Dimensione) e Subnet (Sottorete) o Availability Zone (Zona di disponibilità), scegliere Advanced > > (Avanzate > >) per visualizzare ulteriori opzioni. Di seguito è illustrato il set completo di opzioni:



2. È inoltre possibile sostituire la configurazione predefinita, specificata in gran parte durante la creazione dello stack. Per ulteriori informazioni, consulta Creare un nuovo stack.

Hostname (Nome host)

Identifica l'istanza nella rete. Per impostazione predefinita, AWS OpsWorks Stacks genera il nome host di ogni istanza utilizzando il tema Hostname specificato al momento della creazione dello stack. È possibile sostituire questo valore e specificare il nome host preferito.

Size

Un tipo di EC2 istanza Amazon, che specifica le risorse dell'istanza, come la quantità di memoria o il numero di core virtuali. AWS OpsWorks Stacks specifica una dimensione predefinita per ogni istanza, che puoi sostituire con il tipo di istanza preferito.

I tipi di istanze supportati da AWS OpsWorks Stacks dipendono dal fatto che lo stack si trovi o meno in un VPC. I tipi di istanza sono limitati anche se l'account utilizza il piano gratuito di AWS. L'elenco a discesa Size (Dimensioni) mostra i tipi di istanza supportati per la versione Chef supportata dallo stack. È importante tenere presente che le micro-istanze come t1.micro potrebbero non disporre di risorse sufficienti per supportare alcuni livelli. Per ulteriori informazioni consulta Tipi di istanza .

> **Note**
>
> Se si utilizzano istanze con bilanciamento del carico, ricordare che gli eventi del ciclo di vita Configurazione possono produrre un notevole picco nel carico della CPU, che può durare uno o più minuti. Con le istanze più piccole, questo picco di carico può essere sufficiente per attivare l'upscaling, soprattutto per grandi stack con bilanciamento del carico e con frequenti eventi Configurazione. Di seguito sono elencati alcuni modi per ridurre la probabilità che un evento Configurazione causi un upscaling superfluo.
>
> - Utilizzare istanze più grandi, in modo che il carico aggiuntivo generato da un evento Configurazione non attivi l'upscaling.
> - Non utilizzare tipi di istanza come T2, che condividono risorse CPU.
>
> In questo modo, quando si verifica un evento Configurazione, tutte le risorse CPU dell'istanza sono immediatamente disponibili.
> - Impostare per `exceeded threshold` tempi molto più elevati di quelli necessari per elaborare un evento Configurazione, ad esempio 5 minuti.

Per ulteriori informazioni, consulta Utilizzo del ridimensionamento automatico basato sul carico.

Availability Zone/Subnet (Zona di disponibilità/sottorete)

Se lo stack non è in un VPC, questa impostazione è etichettata come Availability Zone (Zona di disponibilità) ed elenca le zone della regione. È possibile utilizzare questa impostazione per sovrascrivere la zona di disponibilità di default specificata al momento della creazione dello stack.

Se lo stack è in esecuzione in un VPC, questa impostazione è etichettata Subnet (Sottorete) ed elenca le sottoreti del VPC. È possibile utilizzare questa impostazione per sovrascrivere la sottorete di default specificata al momento della creazione dello stack.

> **Note**
>
> Per impostazione predefinita, AWS OpsWorks Stacks elenca gli intervalli CIDR della sottorete. Per rendere l'elenco più leggibile, utilizza la console VPC o l'API per aggiungere un tag a ciascuna sottorete con Key impostato su e Value **`Name`** impostato sul nome della sottorete. AWS OpsWorks Stacks aggiunge quel nome all'intervallo CIDR. Nell'esempio precedente, il tag del nome della sottorete è impostato su **`Private`**.

Scaling Type (Tipo di dimensionamento)

Determina il modo in cui l'istanza viene avviata e arrestata.

- Il valore di default è un'istanza 24/7 ad avvio e arresto manuali.
- AWS OpsWorks Stacks avvia e arresta le istanze basate sul tempo in base a una pianificazione specificata.
- (Solo Linux) AWS OpsWorks Stacks avvia e arresta le istanze basate sul caricamento in base a metriche di carico specificate.

> **Note**
>
> Non è possibile avviare o interrompere personalmente le istanze basate sul tempo o sul carico. Invece, configuri le istanze e AWS OpsWorks Stacks le avvia e le arresta in base alla configurazione. Per ulteriori informazioni, consulta Gestione del carico con istanze basate sul tempo e sul carico.

Chiave SSH

Una coppia di EC2 chiavi Amazon. AWS OpsWorks Stacks installa la chiave pubblica sull'istanza.

- Per le istanze Linux, è possibile utilizzare la chiave privata corrispondente con un client SSH per accedere all'istanza.
- Per le istanze Windows, è possibile utilizzare la chiave privata corrispondente per recuperare la password dell'amministratore dell'istanza. È possibile quindi utilizzare la password con RDP per accedere all'istanza come Amministratore.

Inizialmente, l'impostazione è il valore Default SSH key (Chiave SSH predefinita) specificato durante la creazione dello stack.

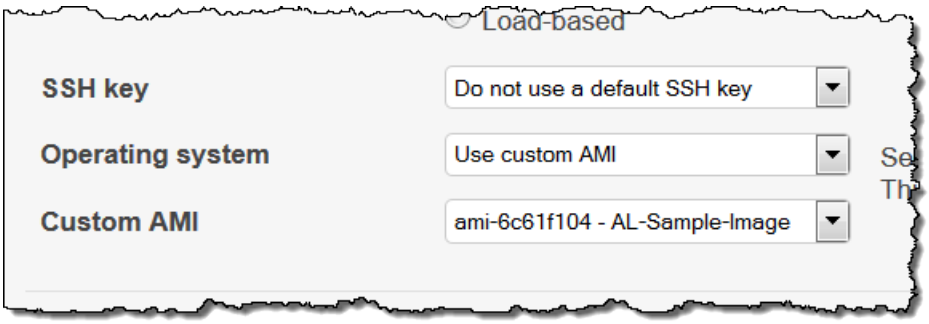
- Se il valore predefinito è impostato su Non utilizzare una chiave SSH predefinita, puoi specificare una delle EC2 chiavi Amazon del tuo account.
- Se il valore predefinito è impostato su una EC2 chiave Amazon, puoi specificare una chiave diversa o nessuna chiave.

Sistema operativo

Il sistema operativo specifica il sistema operativo in esecuzione sull'istanza. AWS OpsWorks Stacks supporta solo sistemi operativi a 64 bit.

Inizialmente, l'impostazione è il valore Default operating system (Sistema operativo predefinito) specificato durante la creazione dello stack. Il valore predefinito può essere sostituito per specificare un altro sistema operativo Linux o un'Amazon Machine Image (AMI) personalizzata. Non è tuttavia possibile passare da Linux a Windows o da Windows a Linux.

Se si seleziona Usa AMI personalizzata, nella pagina viene visualizzato un elenco di tipi di dispositivi personalizzati AMIs anziché Architecture e Root.



Per ulteriori informazioni, consulta Utilizzo di Custom AMIs.

OpsWorks Versione dell'agente

OpsWorks La versione dell'agente specifica la versione dell'agente AWS OpsWorks Stacks che si desidera eseguire sull'istanza. Per permettere a AWS OpsWorks Stacks di aggiornare l'agente automaticamente, scegliere Inherit from stack (Eredita da stack). Per installare una versione specifica dell'agente e aggiornarlo manualmente nell'istanza, scegliere una versione dall'elenco a discesa.

> **Note**
>
> Non tutte le versioni dell'agente funzionano con tutte le release del sistema operativo. Se sull'istanza è in esecuzione un agente, o si installa un agente su un'istanza, che non è completamente supportato dal sistema operativo dell'istanza, la console AWS OpsWorks Stacks visualizza messaggi di errore che richiedono l'installazione di un agente compatibile.

Tenancy

Scegliere l'opzione di tenancy per l'istanza. È possibile scegliere di eseguire le istanze su server fisici totalmente dedicati.

- Default - Rely on VPC settings (Predefinita - Basata su impostazioni VPC). Nessuna tenancy o impostazioni di tenancy ereditate dal VPC.
- Dedicated - Run a dedicated instance (Dedicata - Esegui un'istanza dedicata). È possibile pagare all'ora per le istanze eseguite su un hardware con tenant singolo. Per ulteriori informazioni, consulta le istanze dedicate nella Amazon VPC User Guide e EC2 Amazon Dedicated Instances.

- Dedicated host - Run this instance on a dedicated host (Host dedicato - Avvia questa istanza su un host dedicato). È possibile usufruire di un host fisico a pagamento completamente dedicato all'esecuzione delle istanze, riducendo i costi con le licenze software per socket, per core o per VM esistenti. Per ulteriori informazioni, consulta Panoramica degli host dedicati nella EC2 documentazione di Amazon e Amazon EC2 Dedicated Hosts.

Root device type (Tipo dispositivo root)

Specifica lo storage del dispositivo root dell'istanza.

- Le istanze Linux possono essere supportate da Amazon EBS o supportate da storage di istanze.
- Le istanze Windows devono essere supportate da Amazon EBS.

Per ulteriori informazioni, consulta Storage.

> **Note**
>
> Dopo l'avvio iniziale, le istanze supportate da Amazon EBS si avviano più velocemente delle istanze supportate dall'archivio delle istanze perché AWS OpsWorks Stacks non deve reinstallare il software dell'istanza da zero. Per ulteriori informazioni, consulta Storage del dispositivo root.

Tipo di volume

Specifica il tipo di volume del dispositivo root: Magnetic (Magnetico), Provisioned IOPS (SSD) (IOPS assegnate (SSD)) o General Purpose (SSD) (Per scopi generici (SSD)). Per ulteriori informazioni, consulta la sezione relativa ai tipi di volumi di Amazon EBS.

Volume size (Dimensione dei volumi)

Specifica le dimensioni del volume del dispositivo root per il tipo di volume specificato. Per ulteriori informazioni, consulta la sezione relativa ai tipi di volumi di Amazon EBS.

- General Purpose (SSD) (Per scopi generici (SSD)). Dimensione minima consentita: 8 GiB; dimensione massima 16384 GiB.
- Provisioned IOPS (SSD) (IOPS assegnate (SSD)). Dimensione minima consentita: 8 GiB; dimensione massima 16384 GiB. È possibile impostare un minimo di 100 operazioni input/output al secondo (IOPS) e un massimo di 240 IOPS.

- Magnetico. Dimensione minima consentita: 8 GiB; dimensione massima 1024 GiB.

3. Scegliere Add instance (Aggiungi istanza) per creare la nuova istanza.

> **Note**
>
> Non è possibile sostituire l'impostazione per la versione predefinita dell'agente dello stack al momento della creazione di un'istanza. Per specificare un'impostazione personalizzata per versione dell'agente, è necessario creare l'istanza e quindi modificarne la configurazione.

Per aggiungere un'istanza esistente a un livello

1. Nella pagina Instances (Istanze), scegliere +Instance (+Istanza) per il livello appropriato, quindi aprire la scheda Existing (Esistente).

   > **Note**
   >
   > Se si decide di non usare un'istanza esistente, scegliere New (Nuova) per creare una nuova istanza, come descritto nella procedura precedente.

2. Nella scheda Existing (Esistente), selezionare un'istanza nell'elenco.
3. Scegliere Add instance (Aggiungi istanza) per creare la nuova istanza.

Un'istanza rappresenta un' EC2 istanza Amazon, ma è fondamentalmente solo una struttura dati AWS OpsWorks Stacks. È necessario avviare un'istanza per creare un' EC2 istanza Amazon in esecuzione, come descritto nelle sezioni seguenti.

> **Important**
>
> Se si avviano istanze in un VPC predefinito, è necessario essere cauti nella modifica della configurazione del VPC. Le istanze devono essere sempre in grado di comunicare con il servizio AWS OpsWorks Stacks, Amazon S3 e i repository di pacchetti. Se, ad esempio, rimuovi un gateway predefinito, le istanze perderanno la connessione al servizio AWS OpsWorks Stacks, che quindi considererà le istanze come guaste e le correggerà automaticamente. Tuttavia, AWS OpsWorks Stacks non sarà in grado di installare l'agente di istanza sulle istanze sottoposte a diagnostica automatica. Senza un agente, le istanze non

potranno comunicare con il servizio e il processo di avvio non andrà oltre lo stato `booting`. Per ulteriori informazioni sul VPC predefinito, consulta Piattaforme supportate.

Puoi anche incorporare risorse di calcolo Linux in uno stack creato all'esterno di Stacks: AWS OpsWorks

- EC2 Istanze Amazon create direttamente utilizzando la EC2 console Amazon, la CLI o l'API.
- Istanze locali in esecuzione sul proprio hardware, incluse le istanze in esecuzione su macchine virtuali.

Per ulteriori informazioni, consulta Utilizzo di risorse informatiche create al di fuori di AWS OpsWorks Stacks.

Utilizzo di Custom AMIs

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks supporta due modi per personalizzare le istanze: Amazon Machine Images (AMIs) personalizzate e ricette Chef. Entrambi gli approcci consentono di controllare quali pacchetti e versioni di pacchetti sono installati, come sono configurati e così via. Tuttavia, ogni approccio ha diversi vantaggi, perciò il migliore varia in base alle proprie esigenze.

Di seguito sono elencati i principali motivi per valutare l'utilizzo di un'AMI personalizzata:

- Si desidera raggruppare pacchetti specifici invece di installarli dopo l'avvio dell'istanza.
- È possibile controllare la tempistica degli aggiornamenti del pacchetto per fornire una immagine di base coerente per il livello.
- È opportuno avere istanze, istanze basate sul carico in particolare, per eseguire l'avvio il più rapidamente possibile.

Di seguito sono elencati i principali motivi per valutare l'utilizzo di ricette Chef:

- Sono più flessibili di quelle personalizzate. AMIs
- Sono più facili da aggiornare.
- Possono eseguire gli aggiornamenti sulle istanze in esecuzione.

In pratica, la soluzione ottimale potrebbe essere una combinazione di entrambi gli approcci. Per ulteriori informazioni sulle ricette, consulta Libri di ricette e ricette.

Argomenti

- Come AMIs funziona Custom con AWS OpsWorks Stacks
- Creazione di un'AMI personalizzata per gli AWS OpsWorks stack

Come AMIs funziona Custom con AWS OpsWorks Stacks

Per specificare un AMI personalizzato per le tue istanze, seleziona Usa AMI personalizzato come sistema operativo dell'istanza quando crei una nuova istanza. AWS OpsWorks Stacks visualizza quindi un elenco di quelle personalizzate AMIs nella regione dello stack e tu selezioni quella appropriata dall'elenco. Per ulteriori informazioni, consulta Aggiunta di un'istanza a un livello.

> **Note**
>
> Non è possibile specificare una determinata AMI personalizzata come sistema operativo predefinito dello stack. È possibile impostare `Use custom AMI` come sistema operativo predefinito dello stack, ma è possibile specificare una determinata AMI solo quando si aggiungono nuove istanze ai livelli. Per ulteriori informazioni, consulta Aggiunta di un'istanza a un livello e Creare un nuovo stack. Sebbene sia possibile creare istanze con altri sistemi operativi (come CentOS 6). x) che sono stati creati in modo personalizzato o generato dalla comunità AMIs, non sono supportati ufficialmente.

Questo argomento illustra alcune questioni generali che si devono considerare prima di creare o utilizzare un'AMI personalizzata.

Argomenti

- Comportamento startup

- Scelta di un livello
- Gestione delle applicazioni

Comportamento startup

Quando avvii l'istanza, AWS OpsWorks Stacks utilizza l'AMI personalizzata specificata per avviare una nuova EC2 istanza Amazon. AWS OpsWorks Stacks utilizza quindi cloud-init per installare l'agente AWS OpsWorks Stacks sull'istanza e l'agente esegue le ricette di installazione dell'istanza seguite dalle ricette Deploy. Dopo che l'istanza è online, l'agente esegue le ricette di configurazione per ogni istanza nello stack, tra cui la nuova istanza aggiunta.

Scelta di un livello

L'agente AWS OpsWorks Stacks di solito non è in conflitto con i pacchetti installati. Tuttavia, l'istanza deve essere membro di almeno un livello. AWS OpsWorks Stacks esegue sempre le ricette di quel livello, il che potrebbe causare problemi. È necessario comprendere esattamente ciò che le ricette di un livello fanno a un'istanza prima di aggiungere un'istanza con un'AMI personalizzata a tale livello.

Per vedere quali ricette un determinato tipo di livello esegue su un'istanza, aprire uno stack che include tale livello. Fare clic su Layers (Livelli) nel riquadro di navigazione e fare clic su Recipes (Ricette) per il livello di interesse. Per visualizzare il codice effettivo, fare clic sul nome della ricetta.

> **Note**
>
> Per Linux AMIs, un modo per ridurre la possibilità di conflitti consiste nell'utilizzare AWS OpsWorks Stacks per effettuare il provisioning e configurare l'istanza che è alla base dell'AMI personalizzata. Per ulteriori informazioni, consulta Crea un'AMI Linux personalizzata da un' AWS OpsWorks istanza Stacks.

Gestione delle applicazioni

Oltre ai pacchetti, è possibile includere un'applicazione nell'AMI. Se si dispone di un'applicazione complessa di grandi dimensioni, l'inclusione nell'AMI può ridurre il tempo di avvio dell'istanza. Puoi includere piccole applicazioni nella tua AMI, ma di solito il vantaggio in termini di tempo rispetto alla distribuzione dell'applicazione da parte di AWS OpsWorks Stacks è minimo o nullo.

Un'opzione è di includere l'applicazione nell'AMI e anche creare un'applicazione che distribuisce l'applicazione per le istanze da un archivio. Grazie a questo approccio il tempo di avvio viene

ridotto, ma fornisce anche un modo favorevole per aggiornare l'applicazione dopo che l'istanza è in esecuzione. Si noti che le configurazioni Chef sono idempotenti, perciò le ricette di distribuzione non modificheranno l'applicazione purché la versione nell'archivio sia la stessa di quella nell'istanza.

Creazione di un'AMI personalizzata per gli AWS OpsWorks stack

Per utilizzare un'AMI personalizzata con AWS OpsWorks Stacks, devi prima creare un'AMI da un'istanza personalizzata. Puoi scegliere tra due opzioni:

- Usa la EC2 console o l'API Amazon per creare e personalizzare un'istanza, in base a una versione a 64 bit di uno degli AWS OpsWorks stack supportati AMIs.
- Per Linux AMIs, usa OpsWorks per creare un' EC2 istanza Amazon, in base alla configurazione dei livelli associati.

Prima di creare un'AMI Linux personalizzata, disattivala `noexec` sulla `/tmp` partizione per consentire a AWS OpsWorks Stacks di installare il suo agente su istanze Linux personalizzate.

> **Note**
>
> Si noti che un'AMI potrebbe non funzionare con tutti i tipi di istanze, quindi fare in modo che l'AMI di partenza sia compatibile con i tipi di istanza che si intende utilizzare. In particolare, i tipi di istanze R3 richiedono un'AMI di virtualizzazione basata su hardware (HVM).

Utilizza quindi la EC2 console o l'API Amazon per creare un'AMI personalizzata dall'istanza personalizzata. Puoi usare la tua personalizzazione AMIs in qualsiasi stack che si trova nella stessa regione aggiungendo un'istanza a un layer e specificando l'AMI personalizzata. Per ulteriori informazioni su come creare un'istanza che utilizza un'AMI personalizzata, consulta Aggiunta di un'istanza a un livello.

> **Note**
>
> Per impostazione predefinita, AWS OpsWorks Stacks installa tutti gli aggiornamenti di Amazon Linux all'avvio, il che fornisce la versione più recente. Inoltre, Amazon Linux rilascia una nuova versione all'incirca ogni sei mesi, questo può comportare modifiche rilevanti. Per impostazione predefinita, le versioni personalizzate AMIs basate su Amazon Linux vengono aggiornate automaticamente alla nuova versione quando viene rilasciata. La pratica consigliata consiste nel bloccare l'AMI personalizzata a una specifica versione di

Amazon Linux che permette di rimandare l'aggiornamento finché non è stata testata la nuova versione. Per ulteriori informazioni, consulta Come blocco l'AMI a una versione specifica?

Argomenti

- Crea un'AMI personalizzata utilizzando Amazon EC2
- Crea un'AMI Linux personalizzata da un' AWS OpsWorks istanza Stacks
- Creare un'AMI di Windows personalizzata

Crea un'AMI personalizzata utilizzando Amazon EC2

Il modo più semplice per creare un'AMI personalizzata, nonché l'unica opzione per Windows AMIs, consiste nell'eseguire l'intera operazione utilizzando la EC2 console o l'API Amazon. Per maggiori dettagli sui seguenti passaggi, consulta Creating Your Own. AMIs

Per creare un'AMI personalizzata utilizzando la EC2 console o l'API Amazon

1. Crea un'istanza utilizzando una versione a 64 bit di uno degli AWS OpsWorks Stack supportati AMIs.
2. Personalizzare l'istanza dalla Fase 1 mediante la configurazione, l'installazione di pacchetti e così via. Ricordarsi che tutto ciò che viene installato sarà riprodotto su ogni istanza in base all'AMI, perciò non includere le voci che devono essere specifiche per una determinata istanza.
3. Arrestare l'istanza e creare un'AMI personalizzata.

Crea un'AMI Linux personalizzata da un' AWS OpsWorks istanza Stacks

Per utilizzare un'istanza AWS OpsWorks Stacks Linux personalizzata per creare un'AMI, tieni presente che ogni EC2 istanza Amazon creata da OpsWorks include un'identità unica. Se crei un'AMI personalizzata da tale istanza, include tale identità e tutte le istanze basate sull'AMI hanno la stessa identità. Per assicurarsi che le istanze basate sull'AMI personalizzata abbiano un'identità univoca, è necessario rimuovere l'identità dell'istanza personalizzata prima di creare l'AMI.

Per creare un'AMI personalizzata da un' AWS OpsWorks istanza Stacks

1. Creare uno stack di Linux e aggiungere uno o più livelli per definire la configurazione dell'istanza personalizzata. È possibile utilizzare livelli integrati, personalizzati in base alle esigenze, nonché

livelli personalizzati completi. Per ulteriori informazioni, consulta Personalizzazione degli stack AWS OpsWorks.

2. Modifica i livelli e disabilita AutoHealing.
3. Aggiungere un'istanza con la distribuzione Linux preferita al livello o livelli e avviarla. Consigliamo di utilizzare un'istanza supportata da Amazon EBS. Apri la pagina dei dettagli dell'istanza e registra il suo EC2 ID Amazon per utilizzarlo in un secondo momento.
4. Quando l'istanza è online, accedere con SSH ed eseguire una delle quattro fasi riportate di seguito, a seconda del sistema operativo dell'istanza.
5. Per un'istanza Amazon Linux in uno stack Chef 11 o Chef 12 o un'istanza Red Hat Enterprise Linux 7 in uno stack Chef 11, procedere come segue.

   a. `sudo /etc/init.d/monit stop`
   b. `sudo /etc/init.d/opsworks-agent stop`
   c. `sudo rm -rf /etc/aws/opsworks/ /opt/aws/opsworks/ /var/log/aws/opsworks/ /var/lib/aws/opsworks/ /etc/monit.d/opsworks-agent.monitrc /etc/monit/conf.d/opsworks-agent.monitrc /var/lib/cloud/ /etc/chef`

      > **Note**
      >
      > Per le istanze in uno stack Chef 12, aggiungere le seguenti due cartelle a questo comando:
      >
      > - `/var/chef`
      > - `/opt/chef`

   d. `sudo rpm -e opsworks-agent-ruby`
   e. `sudo rpm -e chef`
6. Per un'istanza Ubuntu 16.04 LTS o 18.04 LTS in uno stack Chef 12, procedere come segue.

   a. `sudo systemctl stop opsworks-agent`
   b. `sudo rm -rf /etc/aws/opsworks/ /opt/aws/opsworks/ /var/log/aws/opsworks/ /var/lib/aws/opsworks/ /etc/monit.d/opsworks-agent.monitrc /etc/monit/conf.d/opsworks-agent.monitrc /var/lib/cloud/ /var/chef /opt/chef /etc/chef`
   c. `sudo apt-get -y remove chef`

d. `sudo dpkg -r opsworks-agent-ruby`

e. `systemctl stop apt-daily.timer`

f. `systemctl stop apt-daily-upgrade.timer`

g. `rm /var/lib/systemd/timers/stamp-apt-daily.timer`

h. `rm /var/lib/systemd/timers/stamp-apt-daily-upgrade.timer`

7. Per altre versioni Ubuntu supportate in uno stack Chef 12, procedere come segue.

   a. `sudo /etc/init.d/monit stop`

   b. `sudo /etc/init.d/opsworks-agent stop`

   c. `sudo rm -rf /etc/aws/opsworks/ /opt/aws/opsworks/ /var/log/aws/opsworks/ /var/lib/aws/opsworks/ /etc/monit.d/opsworks-agent.monitrc /etc/monit/conf.d/opsworks-agent.monitrc /var/lib/cloud/ /var/chef /opt/chef /etc/chef`

   d. `sudo apt-get -y remove chef`

   e. `sudo dpkg -r opsworks-agent-ruby`

8. Per un'istanza Red Hat Enterprise Linux 7 in uno stack Chef 12, procedere come segue.

   a. `sudo systemctl stop opsworks-agent`

   b. `sudo rm -rf /etc/aws/opsworks/ /opt/aws/opsworks/ /var/log/aws/opsworks/ /var/lib/aws/opsworks/ /etc/monit.d/opsworks-agent.monitrc /etc/monit/conf.d/opsworks-agent.monitrc /var/lib/cloud/ /etc/chef /var/chef`

   c. `sudo rpm -e opsworks-agent-ruby`

   d. `sudo rpm -e chef`

9. Questa fase dipende dal tipo di istanza:

   - Per un'istanza supportata da Amazon EBS, usa la console AWS OpsWorks Stacks per arrestare l'istanza e creare l'AMI come descritto in Creazione di un'AMI Linux supportata da Amazon EBS.
   - Per un'istanza supportata dall'archivio di istanze, crea l'AMI come descritto in Creazione di un'AMI Linux supportata dall'archivio di istanze, quindi usa AWS OpsWorks la console Stacks per arrestare l'istanza.

Quando si crea l'AMI, assicurarsi di includere i file del certificato. Ad esempio, è possibile chiamare il comando `ec2-bundle-vol` con l'argomento `-i` impostato su `-i $(find / etc /usr /opt -name '*.pem' -o -name '*.crt' -o -name '*.gpg' | tr '\n' ',')`. Non rimuovere le chiavi pubbliche apt durante il raggruppamento Il comando `ec2-bundle-vol` predefinito gestisce questa attività.

10. Pulisci lo stack tornando alla console AWS OpsWorks Stacks ed eliminando l'istanza dallo stack.

Creare un'AMI di Windows personalizzata

Le seguenti procedure vengono create su misura AMIs per Windows Server 2022 Base. Puoi scegliere altri sistemi operativi Windows Server nella console di EC2 gestione Amazon.

> **Important**
>
> Attualmente, l'agente AWS OpsWorks Stacks non può essere installato e AWS OpsWorks Stacks non può gestire istanze basate su Windows che utilizzano una lingua dell'interfaccia utente di sistema diversa dall'inglese - Stati Uniti d'America (en-US).

Argomenti

- Creazione di un'AMI di Windows personalizzata con Sysprep
- Creazione di un'AMI di Windows personalizzata senza Sysprep
- Aggiunta di una nuova istanza utilizzando un'AMI di Windows personalizzata

Creazione di un'AMI di Windows personalizzata con **`Sysprep`**

La creazione di Windows personalizzate AMIs utilizzando Sysprep comporta in genere un avvio più lento dell'istanza, ma un processo più pulito. Il primo avvio di un'istanza creata da un'immagine creata con `Sysprep` richiede più tempo a causa delle `Sysprep` attività, dei riavvii, del provisioning degli stack e della prima esecuzione AWS OpsWorks degli stack, incluse l'installazione e la configurazione AWS OpsWorks . Completa i passaggi per creare un'AMI Windows personalizzata nella EC2 console Amazon.

Per creare un'AMI personalizzata di Windows con Sysprep

1. Nella EC2 console Amazon, scegli Launch Instance.

2. Trova Microsoft Windows Server 2022 Base, quindi scegli Seleziona.
3. Selezionare il tipo di istanza e selezionare Configure Instance Details (Configura i dettagli dell'istanza). Apportare modifiche di configurazione all'AMI, tra cui il nome della macchina, lo storage e le impostazioni del gruppo di sicurezza. Scegli Avvia.
4. Dopo che il processo di avvio dell'istanza è terminato, ottenere la password e connettersi all'istanza in una finestra Windows Remote Desktop Connection (Connessione desktop remota).
5. Nella schermata Start di Windows, scegli Start, quindi inizia a digitare **`ec2configservice`** finché i risultati non vengono visualizzati nella EC2ConfigServiceSettingsconsole. Apri la console.
6. Nella scheda Generale, assicurati che la casella di controllo Abilita UserData esecuzione sia compilata (sebbene questa opzione non sia richiesta`Sysprep`, è necessaria per AWS OpsWorks Stacks per installare il relativo agente). Deselezionare la casella di controllo per l'opzione Set the computer name of the instance... (Imposta il nome del computer dell'istanza...), perché questa opzione può causare un ciclo di riavvio con AWS OpsWorks Stacks.
7. Nella scheda Immagine, imposta la password dell'amministratore su Casuale per consentire EC2 ad Amazon di generare automaticamente una password che puoi recuperare con una chiave SSH o su Specify per specificare la tua password. `Sysprep`salva questa impostazione. Se si specifica la propria password, archiviare la password in una posizione favorevole. Raccomandiamo di non selezionare Keep Existing (Mantieni esistente).
8. Selezionare Apply (Applica) e Shutdown with Sysprep (Arresta con Sysprep). Quando viene chiesto di confermare, selezionare Yes (Sì).
9. Dopo l'arresto dell'istanza, nella EC2 console Amazon, fai clic con il pulsante destro del mouse sull'istanza nell'elenco Istanze, scegli Immagine, quindi scegli Crea immagine.
10. Nella pagina Create Image (Crea immagine), fornire un nome e una descrizione per l'immagine e specificare la configurazione del volume. Al termine, selezionare Create Image (Crea immagine).
11. Aprire la pagina Images (Immagini) e aspettare che l'immagine passi dalla fase pending (in attesa) a available (disponibile). La nuova AMI è pronta per l'uso.

Creazione di un'AMI di Windows personalizzata senza **`Sysprep`**

Completa i passaggi per creare un'AMI Windows personalizzata nella EC2 console Amazon.

Per creare un'AMI personalizzata di Windows senza Sysprep

1. Nella EC2 console Amazon, scegli Launch Instance.

2. Trova Microsoft Windows Server 2022 Base, quindi scegli Seleziona.
3. Selezionare il tipo di istanza e selezionare Configure Instance Details (Configura i dettagli dell'istanza). Apportare modifiche di configurazione all'AMI, tra cui il nome della macchina, lo storage e le impostazioni del gruppo di sicurezza. Scegli Avvia.
4. Dopo che il processo di avvio dell'istanza è terminato, ottenere la password e connettersi all'istanza in una finestra Windows Remote Desktop Connection (Connessione desktop remota).
5. Nella istanza, aprire `C:\Program Files\Amazon\Ec2ConfigService\Settings\config.xml`, modificare le seguenti due impostazioni e quindi salvare e chiudere il file:
   - `Ec2SetPassword` Da a `Enabled`
   - `Ec2HandleUserData` Da a `Enabled`
6. Disconnettiti dalla sessione di Desktop remoto e torna alla EC2 console Amazon.
7. Nell'elenco Instances (Istanze), arrestare l'istanza.
8. Dopo l'arresto dell'istanza, nella EC2 console Amazon, fai clic con il pulsante destro del mouse sull'istanza nell'elenco Istanze, scegli Immagine, quindi scegli Crea immagine.
9. Nella pagina Create Image (Crea immagine), fornire un nome e una descrizione per l'immagine e specificare la configurazione del volume. Al termine, selezionare Create Image (Crea immagine).
10. Aprire la pagina Images (Immagini) e aspettare che l'immagine passi dalla fase pending (in attesa) a available (disponibile). La nuova AMI è pronta per l'uso.

Aggiunta di una nuova istanza utilizzando un'AMI di Windows personalizzata

Dopo che l'immagine passa allo stato available (disponibile), è possibile creare nuove istanze che si basano sull'AMI di Windows personalizzata. Quando scegli Usa AMI Windows personalizzata dall'elenco dei sistemi operativi, AWS OpsWorks Stacks visualizza un elenco di AMI personalizzati AMIs.

Per aggiungere una nuova istanza basata su un'AMI di Windows personalizzata

1. Quando la tua nuova AMI è disponibile, vai alla console AWS OpsWorks Stacks, apri la pagina Istanze per uno stack di Windows e scegli + Istanza nella parte inferiore della pagina per aggiungere una nuova istanza.
2. Nella scheda New (Nuovo), scegliere Advanced (Avanzate).
3. Nell'elenco a discesa Operating system (Sistema operativo), selezionare Use custom Windows AMI (Usa AMI Windows personalizzata).

4. Nell'elenco a discesa Custom AMI (AMI personalizzata), selezionare l'AMI creata e selezionare Add Instance (Aggiungi istanza).

È ora possibile avviare ed eseguire l'istanza.

Avvio, arresto e riavvio manuali delle istanze 24 ore su 24, 7 giorni su 7

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Puoi utilizzare le istanze 24 ore su 24, 7 giorni su 7 con gli stack sia Linux e che Windows.

Dopo aver aggiunto un'istanza 24 ore su 24, 7 giorni su 7 a un layer, devi avviare manualmente l'istanza per avviare l'istanza Amazon Elastic Compute Cloud EC2 (Amazon) corrispondente e interromperla manualmente per terminare l'istanza Amazon. EC2 Puoi anche riavviare manualmente le istanze che non funzionano correttamente. AWS OpsWorks Stacks avvia e arresta automaticamente le istanze basate sul tempo e sul carico. Per ulteriori informazioni, consulta Gestione del carico con istanze basate sul tempo e sul carico.

> **Important**
>
> AWS OpsWorks Le istanze Stacks devono essere avviate, interrotte e riavviate solo nella console. AWS OpsWorks AWS OpsWorks non riconosce le operazioni di avvio, arresto o riavvio eseguite nella EC2 console Amazon.

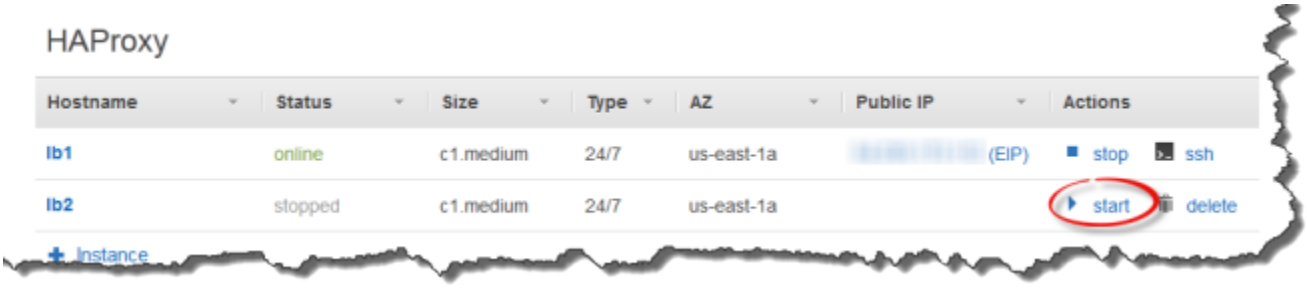
Argomenti

- Avvio o riavvio di un'istanza

- Arresto di un'istanza
- Riavvio di un'istanza

Avvio o riavvio di un'istanza

Per avviare una nuova istanza, nella pagina Instances (Istanze) fai clic su start (avvio) nella colonna Actions (Operazioni).



Puoi anche creare più istanze e quindi avviarle tutte nello stesso momento, fare clic su Start all Instances (Avvia tutte le istanze).

Dopo aver avviato l'istanza, AWS OpsWorks Stacks avvia EC2 un'istanza Amazon e avvia il sistema operativo. Il processo di avvio di solito richiede alcuni minuti ed è in genere leggermente più lento per le istanze Windows rispetto alle istanze Linux. Durante il processo di avvio, nel campo Status (Stato) dell'istanza viene visualizzata la seguente serie di valori:

1. richiesto: AWS OpsWorks Stacks ha chiamato il EC2 servizio Amazon per creare l' EC2 istanza Amazon.
2. in sospeso: AWS OpsWorks Stacks è in attesa dell'avvio dell' EC2istanza Amazon.
3. avvio: l' EC2 istanza Amazon si sta avviando.
4. running_setup - AWS OpsWorks Stacks ha attivato l'evento Setup e sta eseguendo il layer Setup seguite dalle corrispondenti ricette Deploy personalizzate. Per ulteriori informazioni, consulta Esecuzione di ricette. Se hai aggiunto libri di cucina personalizzati allo stack, AWS OpsWorks Stacks installa la versione corrente dal tuo repository prima di eseguire il Setup e Deploy personalizzate.
5. online: l'istanza è pronta per l'uso.

Quando nel campo Status (Stato) viene visualizzato online, significa che l'istanza è operativa.

- Se al layer è collegato un load balancer, AWS OpsWorks Stacks vi aggiunge l'istanza.

- AWS OpsWorks Stacks attiva un Configure , che per ogni istanza esegue le ricette Configure personalizzate.

  Se necessario, queste ricette aggiornare l'istanza in base alla nuova istanza.
- AWS OpsWorks Stacks sostituisce l'azione di avvio dell'istanza con stop, che puoi utilizzare per arrestare l'istanza.

Se l'istanza non è stata avviata correttamente oppure se le ricette di configurazione hanno avuto esito negativo, lo stato verrà impostato rispettivamente su start_failed (avvio non riuscito) o setup_failed (configurazione non riuscita). Puoi esaminare i log per determinarne la causa. Per ulteriori informazioni, consulta Guida al debugging e alla risoluzione dei problemi.

Un'istanza arrestata rimane parte dello stack e conserva tutte le risorse. Ad esempio, i volumi Amazon EBS e gli indirizzi IP elastici sono ancora associati a un'istanza interrotta. Puoi riavviare un'istanza arrestata scegliendo start (avvio) nella colonna Actions (Operazioni) dell'istanza. Il riavvio di un'istanza arrestata comporta quanto segue:

- Istanze supportate da storage di istanze: AWS OpsWorks Stacks lancia una nuova istanza EC2 Amazon con la stessa configurazione.
- Istanze supportate da Amazon EBS: AWS OpsWorks Stacks riavvia l' EC2 istanza Amazon, che ricollega il volume root.

Al termine dell'avvio dell'istanza, Stacks installa gli aggiornamenti del sistema operativo ed esegue AWS OpsWorks Setup e Deploy ricette, proprio come con l'inizio iniziale. AWS OpsWorks Stacks esegue anche le seguenti operazioni per le istanze riavviate, a seconda dei casi.

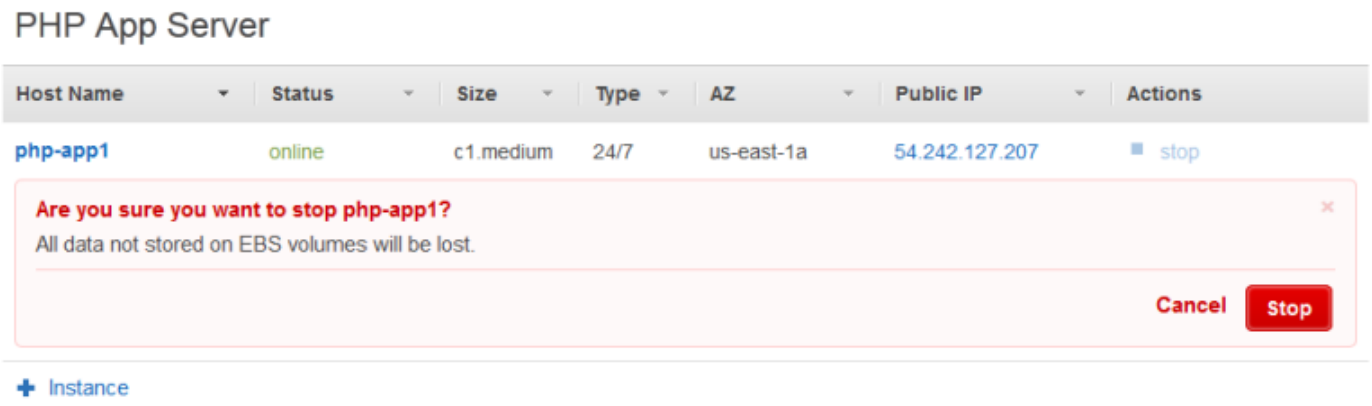
- Riassocia gli indirizzi IP elastici.
- Ricollega i volumi di Amazon Elastic Block Store (Amazon EBS).
- Per le istanze basate su instance store, installa le versioni più recenti dei libri di ricette.

  Le istanze supportate da Amazon EBS continuano a utilizzare i ricettari personalizzati archiviati nel volume root. Se i libri di ricette personalizzati sono stati modificati dopo l'arresto dell'istanza, dovrai aggiornarli manualmente non appena l'istanza è online. Per ulteriori informazioni, consulta Aggiornamento dei libri di ricette personalizzati.

> **Note**
>
> La riassociazione dell'indirizzo IP elastico a un'istanza riavviata potrebbe richiedere alcuni minuti. Considera che l'impostazione Elastic IP (IP elastico) dell'istanza rappresenta i metadati e indica solo che l'indirizzo deve essere associato all'istanza. L'impostazione Public IP (IP pubblico) riflette inoltre lo stato dell'istanza e inizialmente potrebbe essere vuoto. Quando l'indirizzo IP elastico è associato all'istanza, all'indirizzo viene assegnato all'impostazione Public IP (IP pubblico), seguita da (EIP).

Arresto di un'istanza

Nella pagina Istanze, fai clic su Stop nella colonna Azioni dell'istanza, che notifica a AWS OpsWorks Stacks di eseguire le ricette di chiusura e terminare l'istanza. EC2



Puoi anche arrestare ogni istanza nello stack facendo clic su Stop All Instances (Arresta tutte le istanze).

Dopo aver interrotto l'istanza, Stacks esegue diverse attività: AWS OpsWorks

1. Se al layer dell'istanza è collegato un sistema di bilanciamento del carico Elastic Load Balancing, AWS OpsWorks Stacks annulla la registrazione dell'istanza.

   Se il livello supporta la caratteristica Connection Draining del sistema di bilanciamento del carico, AWS OpsWorks Stacks ritarda l'attivazione dell'evento Shutdown evento fino al completamento dell'esaurimento della connessione. Per ulteriori informazioni, consulta Elastic Load Balancing Layer.
2. AWS OpsWorks Stacks attiva un Shutdown , che per l'istanza esegue le ricette Shutdown personalizzate.

3. Dopo l'attivazione dell'evento Shutdown evento, AWS OpsWorks Stacks attende un periodo di tempo specificato per consentire il Shutdown e quindi esegue le seguenti operazioni:
   - Termina le istanze supportate da instance store. Questa operazione elimina tutti i dati.
   - Blocca le istanze supportate da Amazon EBS, che preservano i dati nel volume root.

   Per ulteriori informazioni sullo storage dell'istanza, consulta Storage.

> **Note**
>
> L'impostazione predefinita del timeout di arresto è pari a 120 secondi. Se le ricette di Shutdown richiedono più tempo, puoi modificare la configurazione del livello e modificare l'impostazione.

Puoi monitorare il processo di arresto facendo riferimento alla colonna Status (Stato) dell'istanza. Durante il processo di arresto in tale colonna vengono visualizzati i seguenti valori:

1. terminazione: AWS OpsWorks Stacks sta terminando l'istanza Amazon. EC2
2. shutting_down - Stacks sta eseguendo il livello AWS OpsWorks Shutdown personalizzate.
3. terminata: l' EC2 istanza Amazon viene terminata.
4. stopped (arrestata): l'istanza è stata arrestata.

Riavvio di un'istanza

Nella pagina Instances (Istanze), fai clic sul nome dell'istanza che non funziona per aprire la pagina dei dettagli, quindi fai clic su Reboot (Riavvia).



Questo comando esegue un riavvio graduale dell' EC2 istanza Amazon associata. Non elimina i dati dell'istanza, nemmeno per le istanze supportate da instance store, e non attiva eventi del ciclo di vita.

> **Note**
>
> Per fare in modo che AWS OpsWorks gli stack sostituiscano automaticamente le istanze non riuscite, abilitate la riparazione automatica. Per ulteriori informazioni, consulta Utilizzo della diagnostica automatica.

Gestione del carico con istanze basate sul tempo e sul carico

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Poiché il traffico in entrata può variare, lo stack può disporre di un numero insufficiente o di un numero eccessivo di istanze per gestire il carico. Puoi risparmiare tempo e denaro utilizzando istanze basate sull'orario o istanze basate sul carico per aumentare o ridurre automaticamente le istanze di un livello in modo da avere sempre il numero sufficiente di istanze per poter gestire il traffico in entrata in modo adeguato senza dover sostenere i costi relativi a una capacità superflua. Non è necessario monitorare i carichi del server o avviare o arrestare manualmente le istanze. Inoltre, le istanze basate sull'orario e le istanze basate sul carico eseguono automaticamente le operazioni di distribuzione, dimensionamento e bilanciamento delle applicazioni su più zone di disponibilità all'interno di una regione, garantendo la ridondanza e la scalabilità a livello geografico.

Il dimensionamento automatico si basa su due tipi di istanza, che regolano le istanze online di un livello in base a criteri diversi:

- Istanze basate sull'orario

  Consentono a uno stack di gestire i carichi in base a un modello prevedibile includendo le istanze che vengono eseguite alle ore o nei giorni specificati. Ad esempio, puoi avviare alcune istanze dopo le 18 per eseguire operazioni di backup di particolare entità oppure arrestare alcune istanze durante il fine settimana quando il traffico è meno intenso.
- Istanze basate sul carico

Consentono a uno stack di gestire carichi variabili avviando istanze aggiuntive quando il traffico è elevato e arrestando istanze quando il traffico è meno intenso, in base a vari parametri relativi al carico. Ad esempio, puoi fare in modo che AWS OpsWorks Stacks avvii le istanze quando l'utilizzo medio della CPU supera l'80% e interrompano le istanze quando il carico medio della CPU scende al di sotto del 60%.

Le istanze basate sull'orario e le istanze basate sul carico sono entrambe supportate per gli stack Linux, mentre per gli stack Windows sono supportate solo le istanze basate sull'orario.

A differenza delle istanze di tipo 24 ore su 24, 7 giorni su 7, che devono essere avviate e arrestate manualmente, non devi avviare o arrestare manualmente le istanze basate sull'orario o quelle basate sul carico. Invece, configuri le istanze e AWS OpsWorks Stacks le avvia o le arresta in base alla loro configurazione. Ad esempio, configuri le istanze basate sul tempo in modo che si avviino e si interrompano in base a una pianificazione specifica. AWS OpsWorks Stacks avvia e arresta quindi le istanze in base a tale configurazione.

In genere è consigliabile utilizzare congiuntamente tutti e tre i tipi di istanza, come descritto di seguito.

- Un set di istanze 24 ore su 24, 7 giorni su 7 per gestire il carico di base. In genere devi semplicemente avviare questo tipo di istanze e lasciare in esecuzione in modo continuativo.
- Un insieme di istanze basate sul tempo, che AWS OpsWorks Stacks avvia e interrompe per gestire variazioni prevedibili del traffico. Ad esempio, se durante l'orario di lavoro il traffico è elevato, puoi configurare le istanze basate sull'orario in modo che vengano avviate e arrestate rispettivamente all'inizio e alla fine dell'orario di lavoro.
- Un set di istanze basate sul carico, che AWS OpsWorks Stacks avvia e interrompe per gestire variazioni di traffico imprevedibili. AWS OpsWorks Stacks le avvia quando il carico si avvicina alla capacità delle istanze degli stack 24 ore su 24, 7 giorni su 7 e basate sul tempo, e le interrompe quando il traffico torna alla normalità.

Per ulteriori informazioni su come usare questi modelli di dimensionamento, consulta Ottimizzazione del numero di server.

> **Note**
>
> Se hai creato app per il livello delle istanze o creato libri di cucina personalizzati, AWS OpsWorks Stacks distribuisce automaticamente la versione più recente sulle istanze basate sul tempo e sul caricamento al primo avvio. Tuttavia, AWS OpsWorks Stacks

non distribuisce necessariamente i libri di cucina più recenti su istanze offline riavviate. Per ulteriori informazioni, consulta Modifica delle app e Aggiornamento dei libri di ricette personalizzati.

Argomenti

- Utilizzo del ridimensionamento automatico basato sul tempo
- Utilizzo del ridimensionamento automatico basato sul carico
- In che modo il ridimensionamento basato sul carico differisce dalla riparazione automatica

Utilizzo del ridimensionamento automatico basato sul tempo

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

La scalabilità basata sul tempo consente di controllare quante istanze un layer deve avere online in determinate ore del giorno o giorni della settimana avviando o arrestando le istanze in base a una pianificazione specifica. AWS OpsWorks Stacks esegue i controlli ogni due minuti e avvia o arresta le istanze secondo necessità. Puoi specificare la pianificazione separatamente per ogni istanza, come segue:

- Ora del giorno. Ad esempio, puoi disporre di un numero di istanze in esecuzione maggiore durante il giorno rispetto al numero di istanze disponibili durante la notte.
- Giorno della settimana. Ad esempio, puoi disporre di un numero di istanze in esecuzione maggiore durante i giorni lavorativi rispetto al numero di istanze disponibili durante il fine settimana.

Note

Non puoi indicare date specifiche.

Argomenti

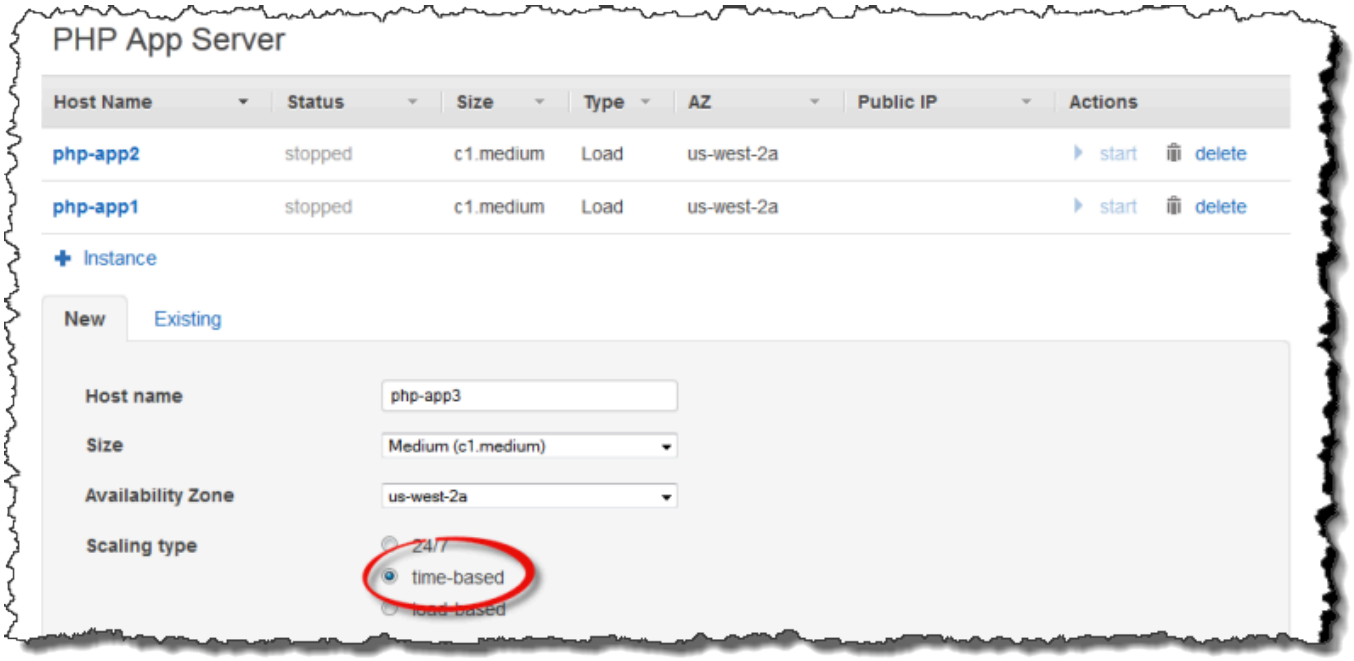
- Aggiungere un'istanza basata sul tempo a un livello
- Configurazione di un'istanza basata sul tempo

Aggiungere un'istanza basata sul tempo a un livello

A un livello puoi aggiungere una nuova istanza basata sull'orario oppure utilizzare un'istanza esistente.

Per aggiungere una nuova istanza basata sull'orario

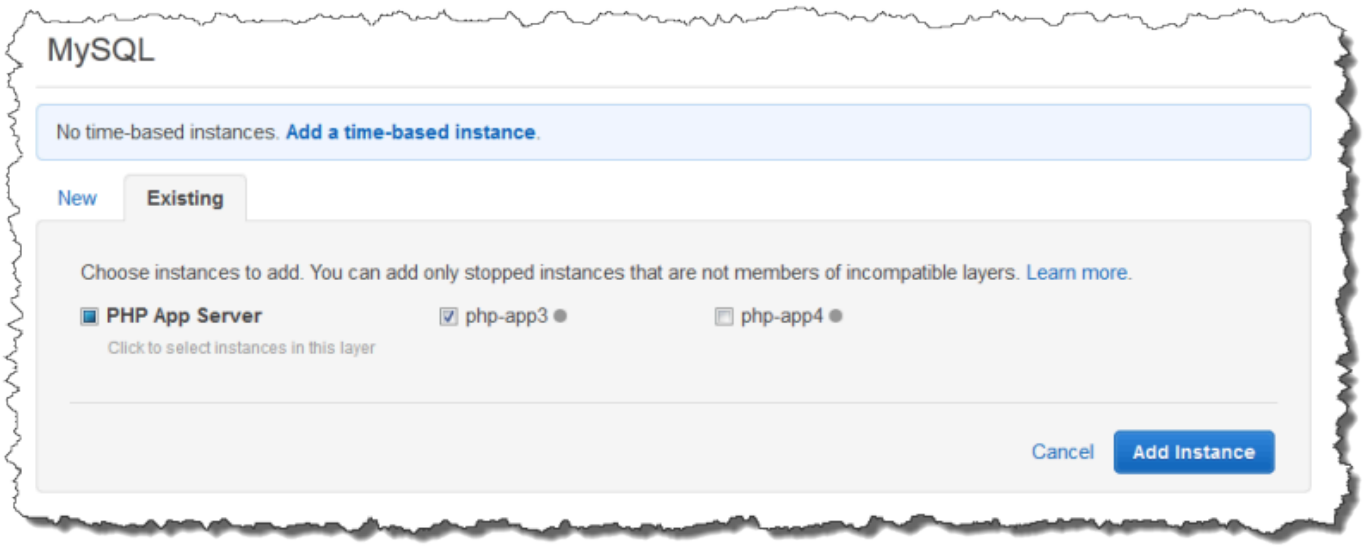
1. Nella pagina Istanze, scegli + Istanza per aggiungere un'istanza. Nella scheda Nuovo, scegli Avanzate, quindi scegli Basato sul tempo.



2. Configura l'istanza. Quindi scegli Aggiungi istanza per aggiungere l'istanza al layer.

Per aggiungere un'istanza esistente basata sull'orario a un livello

1. Nella pagina Istanze basate sul tempo, scegli + Istanza se un layer ha già un'istanza basata sul tempo. Altrimenti, scegli Aggiungi un'istanza basata sul tempo. Quindi scegli la scheda Esistente.



2. Nella scheda Esistente, scegli un'istanza dall'elenco. L'elenco mostra solo le istanze basate sull'orario.

> **Note**
>
> Se cambiate idea sull'utilizzo di un'istanza esistente, nella scheda Nuovo create una nuova istanza, come descritto nella procedura precedente.

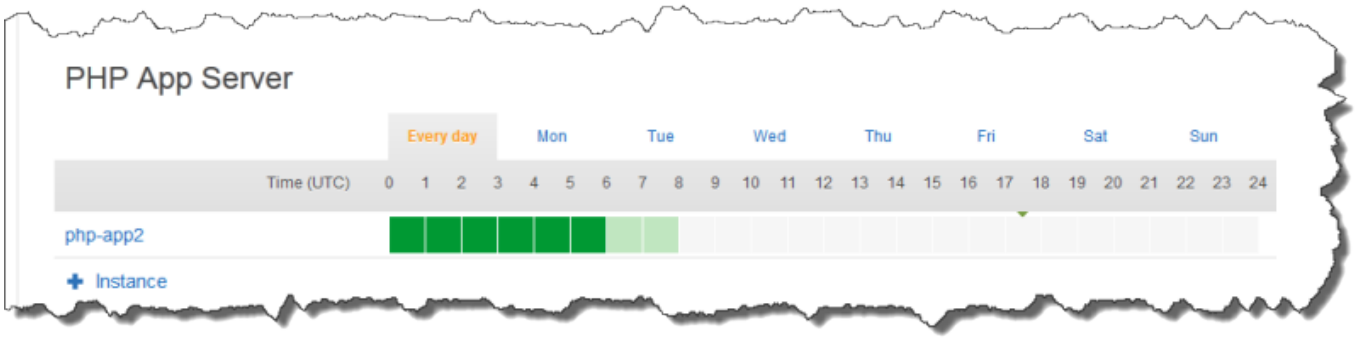
3. Scegliete Aggiungi istanza per aggiungere l'istanza al layer.

Configurazione di un'istanza basata sul tempo

Dopo aver aggiunto un'istanza basata sull'orario a un livello, puoi configurare la relativa pianificazione come segue.

Per configurare un'istanza basata sull'orario

1. Nel riquadro di navigazione, in Istanze, scegli Basato sul tempo.
2. Specificate i periodi online per ogni istanza basata sul tempo compilando le caselle appropriate sotto l'ora desiderata.
   - Per utilizzare la stessa pianificazione ogni giorno, scegli la scheda Tutti i giorni, quindi specifica i periodi di tempo online.
   - Per utilizzare pianificazioni diverse in giorni diversi, scegli ogni giorno, quindi scegli i periodi di tempo appropriati.



> **Note**
>
> Assicurati di prevedere il tempo necessario per avviare un'istanza e che AWS OpsWorks Stacks controlli solo ogni pochi minuti per vedere se le istanze devono essere avviate o interrotte. Ad esempio, se un'istanza deve essere in esecuzione entro 1:00 UTC, avviarla alle 0:00 UTC. In caso contrario, AWS OpsWorks Stacks potrebbe avviare l'istanza solo alcuni minuti dopo l'1:00 UTC e l'istanza impiegherà diversi minuti per essere online.

Puoi modificare i periodi di tempo online di un'istanza in qualsiasi momento eseguendo i passaggi precedenti. Al successivo controllo, AWS OpsWorks Stacks utilizza la nuova pianificazione per determinare se avviare o interrompere le istanze.

> **Note**
>
> Puoi aggiungere una nuova istanza basata sul tempo a un layer aprendo la pagina Time-based e scegliendo Aggiungi un'istanza basata sul tempo (se non hai ancora aggiunto un'istanza basata sul tempo al layer) o + Istanza (se il layer ha già una o più istanze basate sul tempo). Quindi, configura l'istanza come descritto nelle procedure precedenti.

Utilizzo del ridimensionamento automatico basato sul carico

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le istanze basate sul caricamento consentono di avviare o arrestare rapidamente le istanze in risposta alle variazioni del traffico in entrata. AWS OpsWorks Stacks utilizza CloudWatch i dati di Amazon per calcolare le seguenti metriche per ogni livello, che rappresentano i valori medi di tutte le istanze del layer:

- CPU: utilizzo medio della CPU, ad esempio 80%
- Memoria: utilizzo medio della memoria media, ad esempio 60%
- Carico: attività media di calcolo che un sistema esegue in un minuto.

Puoi definire le soglie di upscaling e downscaling per qualsiasi parametro o per tutti i parametri. Puoi anche utilizzare allarmi personalizzati CloudWatch come soglie.

Il superamento di una soglia attiva un evento di dimensionamento. Puoi determinare il modo in cui AWS OpsWorks Stacks risponde all'evento di dimensionamento specificando le seguenti informazioni:

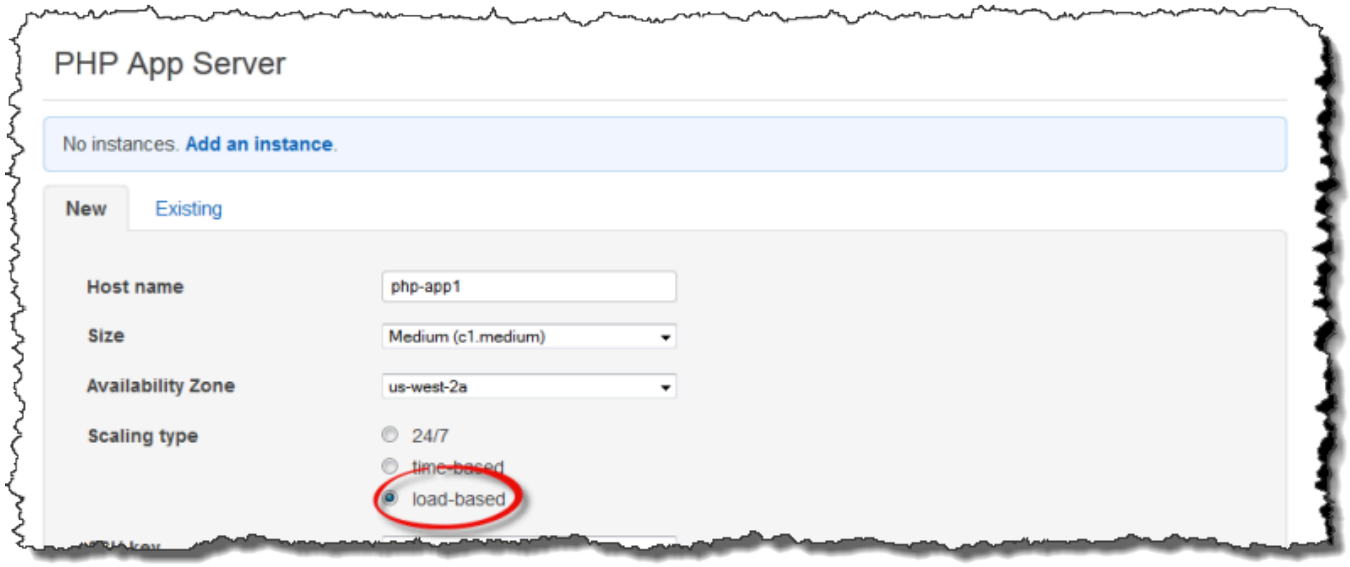
- Numero di istanze da avviare o arrestare.
- Quanto tempo devono attendere AWS OpsWorks gli stack dopo aver superato una soglia prima di avviare o eliminare le istanze. Ad esempio, l'utilizzo della CPU deve superare la soglia di almeno 15 minuti. Questo valore consente di ignorare eventuali variazioni di traffico di breve durata.
- Quanto tempo devono attendere AWS OpsWorks gli Stack dopo l'avvio o l'arresto delle istanze prima di monitorare nuovamente le metriche. In genere devi prevedere un intervallo di tempo sufficiente da consentire alle istanze avviate di essere online o alle istanze arrestate di venire chiuse prima di verificare di nuovo se il livello supera ancora una determinata soglia.

Quando si verifica un evento di scalabilità, AWS OpsWorks Stacks avvia o arresta solo le istanze basate sul carico. Non avvia né arresta le istanze 24 ore su 24, 7 giorni su 7 o le istanze basate sull'orario.

> **Note**
>
> Il dimensionamento automatico basato sul carico non crea nuove istanze, bensì avvia e arresta solo le istanze create. Devi pertanto effettuare il provisioning di un numero sufficiente di istanze basate sul carico in anticipo in modo da gestire il carico massimo previsto.

Per creare un'istanza basata sul carico

1. Nella pagina Istanze, scegli +Instance per aggiungere un'istanza. Scegli Avanzate, quindi scegli in base al caricamento.



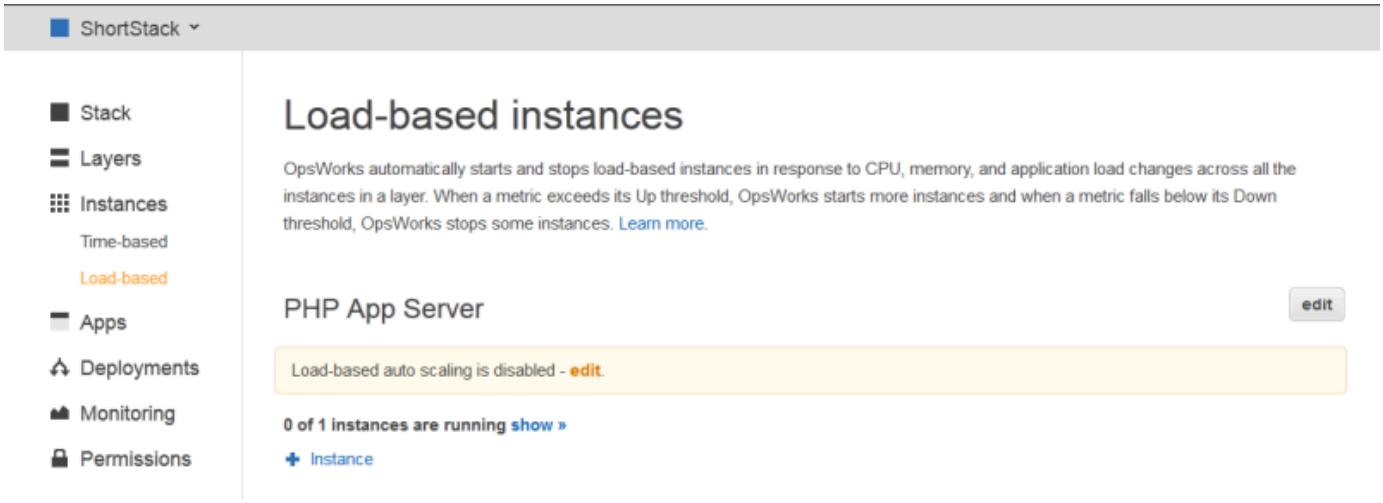
2. Configura l'istanza, quindi scegli Aggiungi istanza per aggiungere l'istanza al layer.

Ripetere questa procedura fino a creare il numero sufficiente di istanze. È possibile aggiungere o rimuovere istanze in un secondo momento, se necessario.

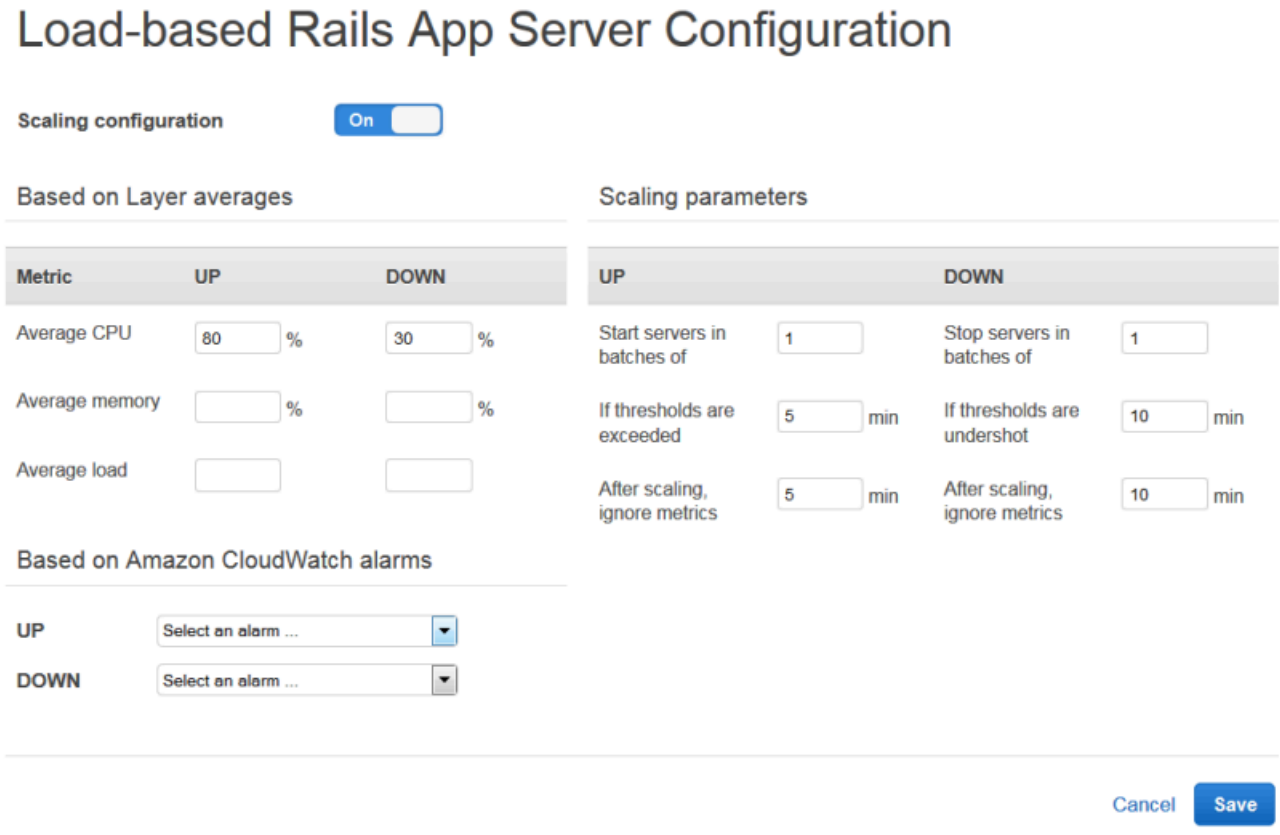
Dopo aver aggiunto le istanze basate sul carico a un livello, è necessario abilitare il dimensionamento basato sul carico e specificare la configurazione. La configurazione del dimensionamento basato sul carico è una proprietà del livello e non una proprietà dell'istanza, che specifica quando un livello deve avviare o arrestare le relative istanze basate sul carico. Questa configurazione deve essere specificata separatamente per ogni livello che utilizza le istanze basate sul carico.

Per abilitare e configurare il dimensionamento automatico basato sul carico

1. Nel riquadro di navigazione, in Istanze, scegli Basato sul caricamento, quindi scegli Modifica per il layer appropriato.



2. Imposta la scalabilità automatica basata sul carico abilitata su On. Impostare quindi i parametri relativi alla soglia e al dimensionamento per definire come e quando aggiungere o eliminare le istanze.



Soglie medie per il livello

Puoi impostare le soglie di dimensionamento in base ai valori seguenti, che rappresentano una media rispetto a tutte le istanze del livello.

- CPU media: utilizzo medio della CPU del layer, come percentuale del totale.
- Memoria media: utilizzo medio della memoria del layer, come percentuale del totale.
- Carico medio: carico medio del layer.

  Per ulteriori informazioni su come viene calcolato il carico, consulta Load (calcolo) su Wikipedia.

Il superamento di una soglia causa un evento di scalabilità, che viene eseguito con upscaling se sono necessarie più istanze e downscaling se sono necessarie meno istanze. AWS OpsWorks Stacks quindi aggiunge o elimina le istanze in base ai parametri di ridimensionamento.

Allarmi personalizzati CloudWatch

Puoi utilizzare fino a cinque CloudWatch allarmi personalizzati come soglie di innalzamento o riduzione. Tali allarmi devono trovarsi nella stessa regione dello stack. Per ulteriori informazioni su come creare allarmi personalizzati, consulta Creazione di allarmi Amazon CloudWatch .

> **Note**
>
> Per usare gli allarmi personalizzati, devi aggiornare il ruolo di servizio in modo da consentire `cloudwatch:DescribeAlarms`. Puoi fare in modo che AWS OpsWorks Stacks aggiorni il ruolo automaticamente la prima volta che usi questa funzionalità oppure puoi modificare il ruolo manualmente. Per ulteriori informazioni, consulta Consentire a AWS OpsWorks Stacks di agire per tuo conto.
> Se sono presenti più allarmi configurati per la configurazione basata sul carico, se un allarme si trova nello stato di allarme `INSUFFICIENT_DATA` metrico, il ridimensionamento dell'istanza basato sul carico non può avvenire anche se è presente un altro allarme. `ALARM` Il ridimensionamento automatico può procedere solo se tutti gli allarmi si trovano nello stato o. `OK` `ALARM` Per ulteriori informazioni sull'uso degli CloudWatch allarmi Amazon, consulta Using Amazon CloudWatch alarms nella Amazon CloudWatch User Guide.

Parametri di dimensionamento

I seguenti parametri controllano il modo in cui AWS OpsWorks Stacks gestisce gli eventi di scalabilità.

- Avvia i server in batch di: il numero di istanze da aggiungere o rimuovere quando si verifica l'evento di scaling.
- Se le soglie vengono superate: il periodo di tempo (in minuti) in cui il carico deve rimanere al di sopra di una soglia di upscaling o al di sotto di una soglia di downscaling prima che Stacks attivi un evento di scaling. AWS OpsWorks
- Dopo il ridimensionamento, ignora le metriche: il periodo di tempo (in minuti) dopo il verificarsi di un evento di ridimensionamento entro il quale Stacks deve ignorare le metriche e sopprimere gli eventi di ridimensionamento aggiuntivi. AWS OpsWorks

  Ad esempio, AWS OpsWorks Stacks aggiunge nuove istanze in seguito a un evento di upscaling, ma le istanze non inizieranno a ridurre il carico finché non saranno state avviate e configurate. Sarebbe inutile generare eventi di dimensionamento aggiuntivi prima che le nuove istanze siano online e gestiscano le richieste. Questa operazione in genere richiede alcuni minuti. Questa impostazione ti consente di indicare a AWS OpsWorks Stacks di eliminare gli eventi di dimensionamento per un periodo sufficiente da permettere alle nuove istanze di risultare online.

  È possibile aumentare questa impostazione per evitare variazioni improvvise di scalabilità quando le medie dei livelli, ad esempio CPU media, memoria media o carico medio, sono temporaneamente discordanti.

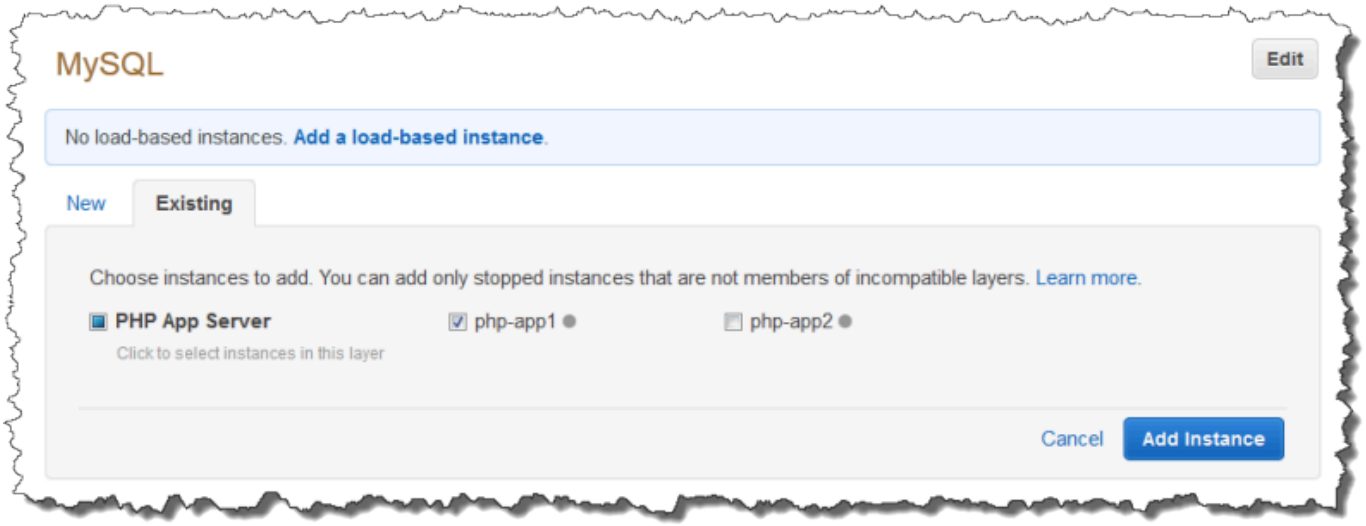
  Ad esempio, se l'utilizzo della CPU è superiore al limite e l'utilizzo della memoria è prossimo al downscaling, un evento di upscale dell'istanza potrebbe essere immediatamente seguito da un evento di downscaling della memoria. Per evitare ciò, puoi aumentare il numero di minuti nell'impostazione After scaling, ignore metrics. In questo esempio, si verificherebbe il ridimensionamento della CPU, ma non l'evento di ridimensionamento della memoria.

3. Per aggiungere altre istanze basate sul caricamento, scegli + Istanza, configura le impostazioni, quindi scegli Aggiungi istanza. Ripetere la procedura fino a creare il numero di istanze basate sul carico sufficiente da gestire il carico massimo previsto. Quindi scegli Save (Salva).

> **Note**
>
> Puoi anche aggiungere una nuova istanza basata sul caricamento a un layer aprendo la pagina Basata sul caricamento e scegliendo Aggiungi un'istanza basata sul caricamento (se non hai ancora aggiunto un'istanza basata sul caricamento al layer) o + Istanza (se il layer ha già una o più istanze basate sul caricamento). Configura quindi l'istanza come descritto nelle procedure precedenti in questa sezione.

Per aggiungere un'istanza esistente basata sul carico a un livello

1. Nel pannello di navigazione, in Istanze, scegli Basato sul caricamento.
2. Se hai già abilitato il ridimensionamento automatico basato sul carico per un layer, scegli + Istanza. Altrimenti, scegli Aggiungi un'istanza basata sul caricamento. Scegli la scheda Esistente.



3. Nella scheda Esistente, scegli un'istanza. L'elenco mostra solo le istanze basate sul carico.

> **Note**
>
> Se cambiate idea sull'utilizzo di un'istanza esistente, nella scheda Nuovo create una nuova istanza come descritto nella procedura precedente.

4. Scegliete Aggiungi istanza per aggiungere l'istanza al layer.

È possibile modificare la configurazione del dimensionamento automatico basato sul carico oppure disabilitarlo in qualsiasi momento.

Per disabilitare il dimensionamento automatico basato sul carico

1. Nel riquadro di navigazione, in Istanze, scegli Basato sul carico, quindi scegli Modifica per il layer appropriato.
2. La scalabilità automatica basata su Switch Load è abilitata su No.

In che modo il ridimensionamento basato sul carico differisce dalla riparazione automatica

Il dimensionamento automatico basato sul carico utilizza i parametri relativi al carico per i quali viene calcolata una media tra tutte le istanze in esecuzione. Se le metriche rimangono tra le soglie specificate, AWS OpsWorks Stacks non avvia né interrompe alcuna istanza. Con la correzione automatica, invece, AWS OpsWorks Stacks avvia automaticamente una nuova istanza con la stessa configurazione quando un'istanza smette di rispondere. L'istanza potrebbe non essere in grado di rispondere a causa di un problema di rete o di un problema al suo interno.

Ad esempio, supponiamo che la soglia di upscaling della CPU sia dell'80% e che un'istanza smetta di rispondere.

- Se la riparazione automatica è disattivata e le restanti istanze in esecuzione riescono a mantenere l'utilizzo medio della CPU al di sotto dell'80%, AWS OpsWorks Stacks non avvia una nuova istanza. Avvierà un'istanza di sostituzione solo se l'utilizzo medio della CPU tra tutte le restanti istanze supera l'80%.
- Se la riparazione automatica è abilitata, AWS OpsWorks Stacks avvia un'istanza sostitutiva indipendentemente dalle soglie di carico.

Utilizzo di risorse informatiche create al di fuori di AWS OpsWorks Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questa funzionalità è supportata solo per stack Linux.

Istanzedescrive come usare AWS OpsWorks Stacks per creare e gestire gruppi di istanze Amazon Elastic Compute Cloud EC2 (Amazon). Puoi anche incorporare risorse informatiche Linux in uno stack creato all'esterno di Stacks: AWS OpsWorks

- EC2 Istanze Amazon create direttamente utilizzando la EC2 console Amazon, la CLI o l'API.
- Istanze locali in esecuzione sul proprio hardware, incluse le istanze in esecuzione su macchine virtuali.

Queste risorse di elaborazione diventano istanze gestite da AWS OpsWorks Stacks e puoi gestirle in modo molto simile alle normali istanze Stacks: AWS OpsWorks

- Gestisci le autorizzazioni degli utenti: puoi utilizzare la gestione degli utenti di AWS OpsWorks Stacks per specificare a quali utenti è consentito accedere ai tuoi stack, quali azioni possono eseguire sulle istanze dello stack e se dispongono di accesso SSH e privilegi sudo.
- Automatizza le attività: puoi fare in modo che AWS OpsWorks Stacks esegua ricette Chef personalizzate per eseguire attività come l'esecuzione di script su una o tutte le istanze di uno stack con un solo comando.

  Se assegni l'istanza a un livello, AWS OpsWorks Stacks esegue automaticamente un set specifico di ricette Chef sull'istanza nei punti chiave del suo ciclo di vita, comprese le ricette personalizzate. Tieni presente che puoi assegnare EC2 istanze Amazon registrate solo a livelli personalizzati.
- Gestisci le risorse: uno stack ti consente di raggruppare e gestire le risorse in un unico Regione AWS stack e la OpsWorks dashboard mostra lo stato dei tuoi stack in tutte le regioni.
- Installa pacchetti: puoi utilizzare le ricette di Chef per installare pacchetti su qualsiasi istanza di uno stack.
- Aggiorna il sistema operativo: AWS OpsWorks Stacks offre un modo semplice per installare patch e aggiornamenti di sicurezza del sistema operativo sulle istanze di uno stack.
- Distribuisci applicazioni: AWS OpsWorks Stacks distribuisce le applicazioni in modo coerente su tutte le istanze del server di applicazioni dello stack.
- Monitoraggio: AWS OpsWorks Stacks crea CloudWatchmetriche personalizzate per monitorare tutte le istanze dello stack.

Per informazioni sui prezzi, consulta la pagina OpsWorks dei prezzi di AWS.

Di seguito è riportato la procedura di base per l'utilizzo di un'istanza registrata.

1. Registrare l'istanza con uno stack.

   L'istanza fa ora parte dello stack ed è gestita da AWS OpsWorks Stacks.
2. Opzionalmente, assegnare l'istanza a un livello.

   Questo passaggio consente di sfruttare appieno la funzionalità di gestione degli AWS OpsWorks Stacks. Puoi assegnare istanze locali registrate a qualsiasi livello; le istanze Amazon EC2 registrate possono essere assegnate solo a livelli personalizzati.
3. Usa AWS OpsWorks Stacks per gestire l'istanza.
4. Quando non ti serve più l'istanza nello stack, annulla la registrazione, così l'istanza viene rimossa da Stacks. AWS OpsWorks

Le seguenti sezioni descrivono in dettaglio questa procedura.

Argomenti

- Registrazione di un'istanza con uno stack Stacks AWS OpsWorks
- Gestione di istanze registrate
- Assegnazione di un'istanza registrata a un livello
- Annullamento dell'assegnazione di un'istanza registrata
- Annullamento della registrazione di un'istanza registrata
- Ciclo di vita dell'istanza registrata

Registrazione di un'istanza con uno stack Stacks AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questa funzionalità è supportata solo per stack Linux.

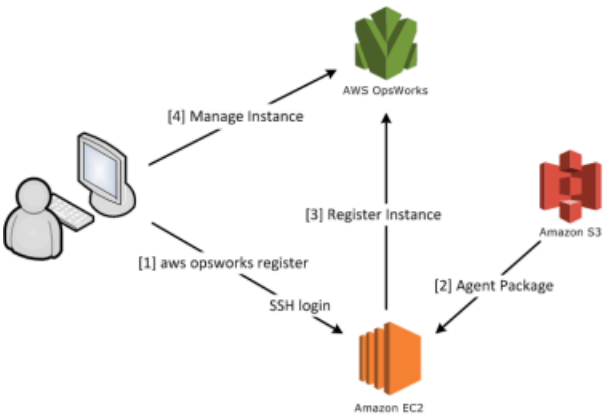
Per registrare un'istanza esterna a AWS OpsWorks Stacks, esegui il comando. AWS CLI aws opsworks register Puoi eseguire il comando dall'istanza che desideri registrare oppure da un altro computer. Applicate `AWSOpsWorksRegisterCLI_OnPremises` le politiche `AWSOpsWorksRegisterCLI_EC2` o a un utente o a un gruppo per concedere le autorizzazioni necessarie rispettivamente AWS CLI per le istanze da registrare EC2 o locali. Queste politiche richiedono la versione 1.16.180 o una versione successiva. AWS CLI

> **Note**
>
> Per impedire agli utenti o ai ruoli di registrare le istanze, aggiorna il profilo dell'istanza per negare l'accesso al comando. register

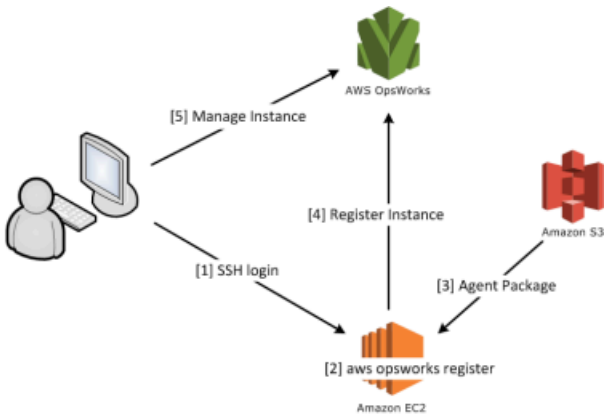
Il processo di registrazione installa un agente su un'istanza che desideri gestire utilizzando AWS OpsWorks Stacks e registra l'istanza con uno stack specificato dall'utente. AWS OpsWorks Dopo la registrazione, l'istanza fa parte dello stack e viene gestita da AWS OpsWorks Stacks. Per ulteriori informazioni, consulta Gestione di istanze registrate.

> **Note**
>
> Sebbene AWS Tools for PowerShell includa il `Register-OpsInstance`cmdlet, che richiama l'azione dell'`register`API, consigliamo invece di utilizzare il AWS CLI per eseguire il `register` comando.

Il diagramma seguente mostra entrambi gli approcci alla registrazione di un'istanza Amazon EC2 . È possibile utilizzare gli stessi approcci per la registrazione di un'istanza locale.



Registering an instance from a workstation



Registering an instance from the instance itself

> **Note**
>
> È possibile utilizzare la console AWS OpsWorks Stacks per gestire un'istanza registrata, ma è necessario eseguire un comando `register` dell'interfaccia a riga di comando di AWS per registrare l'istanza. Il motivo di questo è che il processo di registrazione deve essere eseguito dall'istanza, che non può essere eseguito dalla console.

Le seguenti sezioni descrivono in dettaglio la procedura.

Argomenti

- Scenario: registrare un'istanza dalla workstation
- Registrazione di istanze Amazon EC2 e locali

Scenario: registrare un'istanza dalla workstation

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questa funzionalità è supportata solo per stack Linux.

Il processo di registrazione supporta diversi scenari. Questa sezione illustra un end-to-end esempio di uno scenario: come utilizzare la workstation per registrare un' EC2 istanza Amazon. Altri scenari di registrazione utilizzano una procedura simile. Per ulteriori informazioni, consulta Registrazione di istanze Amazon EC2 e locali.

> **Note**
>
> In genere desideri registrare un' EC2 istanza Amazon esistente. Tuttavia, è possibile creare una nuova istanza e un nuovo stack per l'esempio ed eliminarli al termine.

Argomenti

- Fase 1: creare uno stack e un'istanza
- Fase 2: installa e configura AWS CLI
- Fase 3: Registrare l'istanza con lo EC2 stack di registrazione

Fase 1: creare uno stack e un'istanza

Per iniziare, devi registrare uno stack e un' EC2 istanza Amazon con quello stack.

Per creare lo stack e l'istanza

1. Utilizzare la console AWS OpsWorks Stacks per creare un nuovo stack denominato **`EC2Register`**. È possibile accettare i valori predefiniti per le altre impostazioni dello stack.
2. Avvia una nuova istanza dalla EC2 console Amazon. Tenere presente quanto indicato di seguito.

   - L'istanza deve esistere nella stessa regione e VPC dello stack.

     Se si sta utilizzando un VPC, selezionare una sottorete pubblica per questo scenario.
   - Se è necessario creare una chiave SSH, salvare il file di chiave privata nella workstation e registrare il nome e la posizione del file.

Se si utilizza una chiave esistente, registrare il nome e la posizione del file della chiave privata. Tali valori saranno necessari in seguito.

- L'istanza deve basarsi su uno dei sistemi operativi di Linux supportati. Ad esempio, se il tuo stack si trova negli Stati Uniti occidentali (Oregon), puoi utilizzarlo `ami-35501205` per avviare un'istanza LTS di Ubuntu 14.04 in quella regione.

In caso contrario, accettare i valori predefiniti.

Anche se l'istanza è stata avviata, è possibile procedere alla sezione successiva.

Fase 2: installa e configura AWS CLI

La registrazione viene eseguita utilizzando il comando. AWS CLI aws opsworks register Prima di registrare la prima istanza, è necessario utilizzare la versione 1.16.180 AWS CLI o una versione successiva. I dettagli dell'installazione dipendono dal sistema operativo della workstation. Per ulteriori informazioni sull'installazione di AWS CLI, consulta Installazione dell'interfaccia a riga di comando AWS. Per verificare la versione di AWS CLI in esecuzione, immetti `aws --version` in una sessione shell.

> **Note**
>
> Per impedire agli utenti o ai ruoli di registrare le istanze, aggiorna il profilo dell'istanza per negare l'accesso al comando. register

Ti consigliamo vivamente di non saltare questo passaggio, anche se lo stai già eseguendo sulla AWS CLI tua workstation. L'uso della versione più recente di AWS CLI è una best practice di sicurezza.

È necessario fornire `register` con un set di credenziali AWS che hanno le autorizzazioni appropriate. Il metodo consigliato per eseguire questa operazione, in modo da evitare di installare le credenziali direttamente su un'istanza, consiste nel registrare le istanze avviate con un profilo di istanza e quindi aggiungere lo switch al comando. `--use-instance-profile` `register` Se si ottengono le credenziali da un profilo dell'istanza, passare a Fase 3: Registrare l'istanza con lo EC2 stack di registrazione in questo argomento. Tuttavia, se l'istanza non è stata avviata con un profilo dell'istanza, è possibile creare un utente IAM. La procedura seguente crea un nuovo utente con le autorizzazioni appropriate, installa le credenziali dell'utente sulla workstation e quindi passa tali credenziali a. `register`

> **Warning**
>
> Gli utenti IAM dispongono di credenziali a lungo termine, il che rappresenta un rischio per la sicurezza. Per ridurre questo rischio, si consiglia di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari.

Per creare l'utente

1. Nella console IAM, selezionare Users (Utenti) nel riquadro di navigazione e selezionare Add user (Aggiungi utente).
2. Aggiungere un utente denominato **`EC2Register`**.
3. Scegli Next (Successivo).
4. Nella pagina Imposta autorizzazioni, scegli Allega direttamente le politiche.
5. Entra **`OpsWorks`** nella casella del filtro delle politiche di autorizzazione per visualizzare le AWS OpsWorks politiche, seleziona una delle seguenti politiche, quindi scegli Avanti: revisione. La policy concede all'utente le autorizzazioni necessarie per eseguire `register`.
   - Scegli `AWSOpsWorksRegisterCLI_EC2` di consentire all'utente le autorizzazioni per registrare le EC2 istanze che utilizzano i profili di istanza.
   - Scegliere `AWSOpsWorksRegisterCLI_OnPremises` per concedere all'utente le autorizzazioni per registrare istanze locali.
6. Scegli Next (Successivo).
7. Nella pagina Review (Verifica) scegli Create user (Crea utente).
8. Ora crea le chiavi di accesso per il tuo utente. Dal riquadro di navigazione, scegli Utenti, quindi scegli l'utente per cui desideri creare le chiavi di accesso.
9. Scegli la scheda Credenziali di sicurezza, quindi scegli Crea chiave di accesso.
10. Scegli le best practice e le alternative delle chiavi di accesso più adatte alla tua attività.
11. Scegli Next (Successivo).
12. (Facoltativamente) inserisci un tag per identificare le chiavi di accesso.
13. Scegli Next (Successivo).
14. Scegliete Scarica il file.csv, salvate il file delle credenziali in una posizione comoda sul sistema e scegliete Fine.

È necessario fornire le credenziali utente IAM a `register`. Questa procedura dettagliata gestisce l'operazione installando le credenziali di EC2 registrazione nel file della workstation. `credentials` Per informazioni su altri modi di gestire le credenziali per AWS CLI, vedere File di configurazione e credenziali.

Per installare le credenziali dell'utente

1. Creare o aprire il file `credentials` della workstation. Il file si trova in `~/.aws/credentials` (Linux, Unix e OS X) o `C:\Users\`*`User_Name`*`\.aws\credentials` (sistemi Windows).
2. Aggiungi un profilo per l'utente EC2 Register al `credentials` file, utilizzando il seguente formato.

   ```
   [ec2register]
   aws_access_key_id = access_key_id
   aws_secret_access_key = secret_access_key
   ```

   Sostituisci *`access_key_id`* e *`secret_access_key`* con le chiavi di EC2 registro che hai scaricato in precedenza.

Fase 3: Registrare l'istanza con lo EC2 stack di registrazione

A questo punto è possibile registrare l'istanza.

Per registrare l'istanza

1. In AWS OpsWorks Stacks, torna allo stack di EC2 registrazione, scegli Istanze nel riquadro di navigazione, quindi scegli Registra un'istanza.
2. Seleziona EC2 Istanze, scegli Avanti: Seleziona istanze e seleziona la tua istanza dall'elenco.
3. Scegli Avanti: Installa AWS CLI e Avanti: Registra istanze. AWS OpsWorks Stacks utilizza automaticamente le informazioni disponibili, come l'ID dello stack e l'ID dell'istanza per creare un modello di **`register`** comando, che viene visualizzato nella pagina Registra istanze. In questo esempio, viene usato `register` per accedere all'istanza con una chiave SSH e specificherà esplicitamente il file chiave, quindi impostare I use SSH keys to connect to my instances (Utilizzo chiavi SSH per connettermi alle istanze) su Yes (Sì). Il modello di comando è simile al quanto segue.

   ```
   aws opsworks register --infrastructure-class ec2 --region region endpoint ID
   ```

```
--stack-id 247be7ea-3551-4177-9524-1ff804f453e3 --ssh-username [username]
--ssh-private-key [key-file] i-f1245d10
```

> **Note**
>
> È necessario impostare la regione sulla regione dell'endpoint del servizio AWS OpsWorks Stacks, non sulla regione dello stack, se lo stack si trova all'interno di una regione classica associata all'endpoint regionale. `us-east-1` AWS OpsWorks Stacks determina la regione dello stack dall'ID dello stack.

4. Il modello di comando contiene vari valori di argomento specifici per l'utente, che sono indicati da parentesi e devono essere sostituiti con valori appropriati. Copiare il modello di comando in un editor di testo e modificarlo come segue.

> **Important**
>
> L'utente IAM creato durante il processo di registrazione è necessario per tutta la durata di un'istanza registrata. L'eliminazione dell'utente impedisce all'agente AWS OpsWorks Stacks di comunicare con il servizio. Per evitare problemi nella gestione delle istanze registrate nel caso in cui l'utente venga eliminato accidentalmente, aggiungi il `--use-instance-profile` parametro al `register` comando per utilizzare invece il profilo di istanza integrato dell'istanza. L'aggiunta del `--use-instance-profile` parametro impedisce inoltre che si verifichino errori quando si ruotano le chiavi di accesso AWS dell'account ogni 90 giorni (una best practice consigliata), poiché previene le discrepanze tra le chiavi di accesso disponibili per l' AWS OpsWorks agente e l'utente IAM richiesto.

- Sostituisci *`key file`* con il percorso completo del file di chiave privata per la coppia di EC2 chiavi Amazon che hai salvato quando hai creato l'istanza.

  È possibile utilizzare un percorso relativo, se si preferisce.

- Sostituiscilo *`username`* con il nome utente dell'istanza.

  In questo esempio, il nome utente è `ubuntu`, per un'istanza Ubuntu oppure `ec2-user` per un'istanza Red Hat Enterprise Linux (RHEL) oppure Amazon Linux.

- Aggiungi`--use-instance-profile`, che viene eseguito `register` con il profilo dell'istanza per evitare errori durante la rotazione delle chiavi o se l'utente IAM principale viene eliminato accidentalmente.

Il comando dovrebbe essere simile a quanto segue.

```
aws opsworks register --use-instance-profile --infrastructure-class ec2 \
  --region us-west-2  --stack-id 247be7ea-3551-4177-9524-1ff804f453e3 --ssh-
username ubuntu \
  --ssh-private-key "./keys/mykeys.pem" i-f1245d10
```

5. Aprire la finestra di un terminale sulla workstation, incollare il comando `register` dall'editor ed eseguire il comando.

   La registrazione di solito richiede circa cinque minuti. Al termine, torna alla console AWS OpsWorks Stacks e scegli Fine. Quindi, nel riquadro di navigazione scegliere Instances (Istanze). L'istanza dovrebbe essere elencata sotto Unassigned Instances (Istanze non assegnate). È possibile poi assegnare l'istanza a un livello o lasciarlo dove si trova, a seconda di come si intende gestire l'istanza.

6. Al termine, interrompi l'istanza ed eliminala utilizzando la console o i AWS OpsWorks comandi di Stacks. Ciò interrompe l' EC2istanza Amazon, quindi non dovrai sostenere ulteriori addebiti.

Registrazione di istanze Amazon EC2 e locali

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questa funzionalità è supportata solo per stack Linux.

Questa sezione descrive come registrare un'istanza Amazon EC2 o locale con uno AWS OpsWorks stack Stacks.

Argomenti

- Preparazione dell'istanza
- Installazione e configurazione dell' AWS CLI
- Registrazione dell'istanza
- Utilizzo del comando register
- Comandi register di esempio
- Policy di registrazione delle istanze

Preparazione dell'istanza

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questa funzionalità è supportata solo per stack Linux.

Prima di registrare un'istanza, è necessario accertarsi che sia compatibile con AWS OpsWorks Stacks. I dettagli dipendono dal fatto che tu stia registrando un'istanza locale o Amazon EC2 .

Istanze locali

Un'istanza locale deve soddisfare i seguenti criteri:

- L'istanza deve essere eseguita su uno dei sistemi operativi di Linux supportati. Sebbene sia possibile creare o registrare istanze con altri sistemi operativi (come CentOS 6). x) che sono stati creati in modo personalizzato o generato dalla community AMIs, non sono supportati ufficialmente.

È necessario installare il pacchetto `libyaml` nell'istanza. Per istanze Ubuntu, il pacchetto è denominato `libyaml-0-2`. Per le istanze CentOS e Red Hat Enterprise Linux, il pacchetto è denominato `libyaml`.

- L'istanza deve avere un tipo di istanza supportata (a volte definita come le dimensioni dell'istanza). I tipi di istanza possono variare a seconda del sistema operativo e variano a seconda se lo stack si trova in un VPC. Per un elenco dei tipi di istanze supportati, visualizza i valori dell'elenco a discesa Size che vengono visualizzati nella console AWS OpsWorks Stacks quando tenti di creare una nuova istanza nello stack di destinazione. Se un tipo di istanza è disattivato e non può essere creato nello stack target, non è possibile registrare un'istanza di quel tipo.
- L'istanza deve disporre di un accesso a Internet che le consenta di comunicare con l'endpoint del servizio AWS OpsWorks Stacks,. `opsworks.us-east-1.amazonaws.com (HTTPS)` L'istanza deve inoltre supportare connessioni in uscita a risorse AWS come Amazon S3.
- Se si prevede di registrare l'istanza da un'altra workstation, l'istanza registrata deve supportare l'accesso a SSH dalla workstation.

  L'accesso SSH non è necessario se si esegue il comando di registrazione dall'istanza.
- La chiave di AWS accesso viene utilizzata per l'autenticazione dall' AWS OpsWorks agente al servizio AWS OpsWorks Stacks. Se ruoti le chiavi di accesso come consigliato ogni 90 giorni, aggiorna l' AWS OpsWorks agente manualmente per utilizzare la nuova chiave. Su un computer o un'istanza locale, modifica il `/etc/aws/opsworks/instance-agent.yml` file con la nuova chiave di accesso e la chiave segreta. Il seguente comando mostra la chiave di accesso e la chiave segreta in questo file. Se un agente utilizza chiavi obsolete, questo può causare degli errori.

```
cat /etc/aws/opsworks/instance-agent.yml | egrep "access_key|secret_key"
:access_key_id: AKIAIOSFODNN7EXAMPLE
:secret_access_key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

EC2Istanze Amazon

Un' EC2 istanza Amazon deve soddisfare i seguenti criteri:

- L'AMI deve basarsi su uno dei sistemi operativi di Linux supportati. Per un elenco corrente, consulta AWS OpsWorks Sistemi operativi impilati.

  Per ulteriori informazioni, consulta Utilizzo di Custom AMIs.

Se l'istanza si basa su un'AMI personalizzata che deriva da AMI supportate standard AMI oppure se l'istanza contiene una configurazione proprio minima, è necessario installare il pacchetto `libyaml` sull'istanza. Per istanze Ubuntu, il pacchetto è denominato `libyaml-0-2`. Per le istanze Amazon Linux e Red Hat Enterprise Linux, il pacchetto è denominato`libyaml`.

- L'istanza deve avere un tipo di istanza supportata (a volte definita come le dimensioni dell'istanza). I tipi di istanza possono variare a seconda del sistema operativo e variano a seconda se lo stack si trova in un VPC. Per un elenco dei tipi di istanze supportati, visualizza i valori dell'elenco a discesa Size che vengono visualizzati nella console AWS OpsWorks Stacks quando tenti di creare una nuova istanza nello stack di destinazione. Se un tipo di istanza è disattivato e non può essere creato nello stack target, non è possibile registrare un'istanza di quel tipo.
- L'istanza deve essere nello stato `running`.
- L'istanza non deve far parte di un gruppo di Auto Scaling.

  Per ulteriori informazioni, consulta Scollegare EC2 le istanze dal gruppo di Auto Scaling.
- L'istanza può far parte di un VPC, ma deve trovarsi nello stesso VPC dello stack e il VPC deve essere configurato per funzionare correttamente con gli stack. AWS OpsWorks
- Le istanze Spot non sono supportate, poiché non funzionano con la diagnostica automatica.

Quando registri un' EC2 istanza Amazon, AWS OpsWorks Stacks non modifica i gruppi o le regole di sicurezza dell'istanza. Assicurati che le regole del gruppo di sicurezza dell'istanza soddisfino i seguenti requisiti AWS OpsWorks Stacks.

Regole in ingresso

Le regole in ingresso devono permettere quanto indicato di seguito.

- Accesso SSH.
- Traffico dai livelli appropriati.

  Ad esempio, un server di database in genere consente il traffico in entrata dai livelli server dell'applicazione dello stack.
- Traffico alle porte appropriate.

  Ad esempio, le istanze di server dell'applicazione in genere consentono tutto il traffico in entrata per le porte 80 (HTTP) e 443 (HTTPS).

Regole in uscita

Le regole in uscita devono permettere quanto indicato di seguito.

- Traffico verso il servizio AWS OpsWorks Stacks proveniente dalle applicazioni in esecuzione sull'istanza.
- Traffico per accedere a risorse AWS come Amazon S3 da applicazioni che utilizzano l'API AWS.

Un approccio comune è quello di non specificare alcuna regola in uscita, che non applica alcuna restrizione al traffico in uscita.

Installazione e configurazione dell' AWS CLI

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Prima di registrare la prima istanza, è necessario eseguire la versione 1.16.180 AWS CLI o una versione successiva sul computer da cui si esegue. `register` I dettagli dell'installazione dipendono dal sistema operativo della workstation. Per ulteriori informazioni sull'installazione AWS CLI, consulta Installazione dell'interfaccia a riga di comando AWS e Configurazione dell'interfaccia a riga di comando AWS. Per verificare la versione di AWS CLI in esecuzione, immetti `aws --version` in una sessione shell.

> **Note**
>
> Sebbene AWS Tools for PowerShell includa il `Register-OpsInstance`cmdlet, che richiama l'azione dell'`register`API, consigliamo invece di utilizzare il AWS CLI per eseguire il `register` comando.

È necessario eseguire `register` con le autorizzazioni appropriate. Puoi ottenere le autorizzazioni utilizzando un ruolo IAM o, in modo meno ottimale, installando credenziali utente con le autorizzazioni

appropriate sulla workstation o sull'istanza da registrare. È quindi possibile eseguire `register` con tali credenziali, come descritto più avanti. Specificate le autorizzazioni allegando una policy IAM all'utente o al ruolo. Infatti`register`, utilizzi `AWSOpsWorksRegisterCLI_OnPremises` le politiche `AWSOpsWorksRegisterCLI_EC2` o, che concedono le autorizzazioni per registrare istanze Amazon EC2 o locali, rispettivamente.

> **Note**
>
> Se esegui `register` su un' EC2 istanza Amazon, idealmente dovresti utilizzare un ruolo IAM per fornire le credenziali. Per ulteriori informazioni su come associare un ruolo IAM a un'istanza esistente, consulta Collegare un ruolo IAM a un'istanza o Sostituire un ruolo IAM nella Amazon EC2 User Guide.

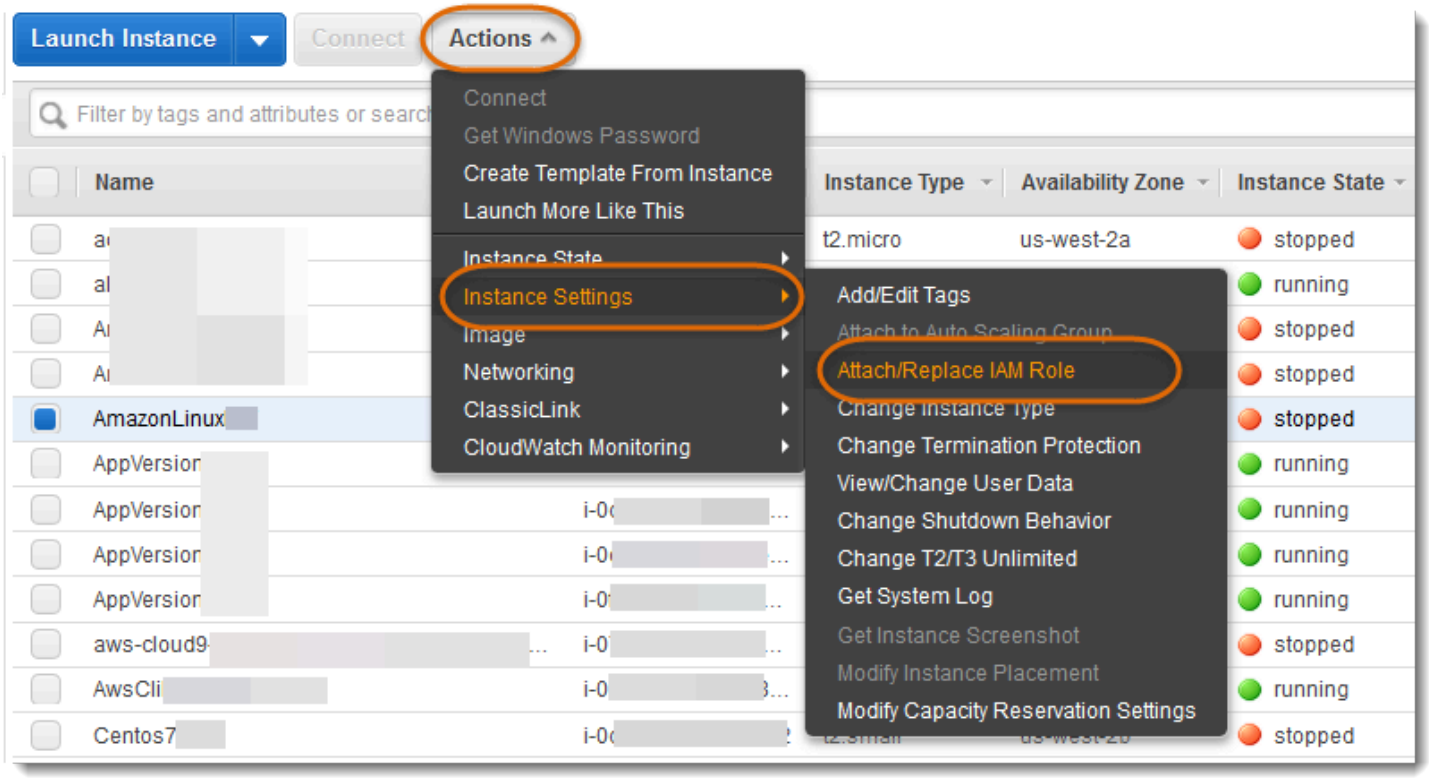
Per frammenti di codice di esempio delle policy `AWSOpsWorksRegisterCLI_EC2` e `AWSOpsWorksRegisterCLI_OnPremises`, consulta Policy di registrazione delle istanze. Per ulteriori informazioni sulla creazione e sulla gestione delle credenziali AWS, consulta Credenziali di sicurezza AWS.

Argomenti

- Utilizzo di un ruolo IAM
- Utilizzo delle credenziali installate

Utilizzo di un ruolo IAM

Se esegui il comando dall' EC2 istanza Amazon che intendi registrare, la strategia preferita per fornire le credenziali `register` consiste nell'utilizzare un ruolo IAM con la `AWSOpsWorksRegisterCLI_EC2` policy o autorizzazioni equivalenti allegate. Questo approccio consente di evitare l'installazione delle credenziali nell'istanza. Un modo per farlo è utilizzare il comando Attach/Replace IAM Role nella EC2 console, come mostrato nell'immagine seguente.



Per ulteriori informazioni su come associare un ruolo IAM a un'istanza esistente, consulta Collegare un ruolo IAM a un'istanza o Sostituire un ruolo IAM nella Amazon EC2 User Guide. Per istanze avviate con un profilo dell'istanza (opzione consigliata), aggiungere il parametro `--use-instance-profile` al comando `register` per fornire le credenziali. Non utilizzare il parametro `--profile`.

Se l'istanza è in esecuzione e ha un ruolo, è possibile concedere le autorizzazioni necessarie collegando la policy `AWSOpsWorksRegisterCLI_EC2` al ruolo. Il ruolo fornisce un set di credenziali predefinite per l'istanza. Se le credenziali non sono state installate nell'istanza, `register` automaticamente assume il ruolo e viene eseguito con le autorizzazioni.

> **Important**
>
> È consigliabile non installare le credenziali sull'istanza. Oltre a creare un rischio per la sicurezza, il ruolo dell'istanza si colloca alla fine della catena di provider predefiniti che AWS CLI utilizza per individuare le credenziali predefinite. Le credenziali installate potrebbero richiedere la precedenza sul ruolo e `register` potrebbe quindi non avere le autorizzazioni necessarie. Per ulteriori informazioni, consulta Nozioni di base di AWS CLI.

Se un'istanza in esecuzione non ha un ruolo, è necessario installare le credenziali con le autorizzazioni necessarie sull'istanza, come descritto in Utilizzo delle credenziali installate. È consigliabile, più semplice e meno soggetto a errori utilizzare le istanze avviate con un profilo dell'istanza.

Utilizzo delle credenziali installate

Esistono diversi modi per installare le credenziali utente su un sistema e fornirle a un AWS CLI comando. Di seguito viene descritto un approccio che non è più consigliato, ma che può essere utilizzato se si registrano EC2 istanze avviate senza un profilo di istanza. È inoltre possibile utilizzare le credenziali esistenti di un utente purché le policy collegate concedano le autorizzazioni necessarie. Per ulteriori informazioni, tra cui una descrizione di altri modi per installare le credenziali, consulta File di configurazione e delle credenziali.

Utilizzare le credenziali installate

1. Crea un utente IAM e salva l'ID della chiave di accesso e la chiave di accesso segreta in una posizione sicura.

   > **Warning**
   >
   > Gli utenti IAM dispongono di credenziali a lungo termine, il che rappresenta un rischio per la sicurezza. Per ridurre questo rischio, si consiglia di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari.

2. Allega la AWSOpsWorksRegisterCLI_OnPremises policy all'utente. Se si preferisce, è possibile collegare una policy che concede autorizzazioni più ampie, purché includa le autorizzazioni `AWSOpsWorksRegisterCLI_OnPremises`.

3. Creare un profilo per l'utente nel file `credentials` del sistema. Il file si trova in `~/.aws/credentials` (Linux, Unix e OS X) o `C:\Users\`*`User_Name`*`\.aws\credentials` (sistemi Windows). Il file contiene uno o più profili nel seguente formato, ognuno dei quali contiene l'ID della chiave di accesso dell'utente e una chiave di accesso segreta.

   ```
   [profile_name]
   aws_access_key_id = access_key_id
   aws_secret_access_key = secret_access_key
   ```

Sostituisci i valori and con le credenziali IAM salvate *access_key_id* in precedenza. *secret_access_key* È possibile specificare qualsiasi nome si preferisce per un nome del profilo, con due limitazioni: il nome deve essere unico e il profilo predefinito deve essere denominato `default`. È inoltre possibile utilizzare un profilo esistente, purché associato alle autorizzazioni necessarie.

4. Utilizzare il parametro `--profile` del comando `register` per specificare il nome del profilo. Il comando `register` verrà eseguito con le autorizzazioni concesse alle credenziali associate.

   È possibile anche omettere `--profile`. In questo caso, `register` viene eseguito con le credenziali predefinite. Si noti che queste non sono necessariamente le credenziali del profilo predefinito, pertanto è necessario accertarsi che le credenziali predefinite abbiano le autorizzazioni necessarie. Per ulteriori informazioni su come AWS CLI determina le credenziali predefinite, consulta Configurazione dell'interfaccia a riga di comando AWS.

Registrazione dell'istanza

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questa funzionalità è supportata solo per stack Linux.

È possibile registrare un'istanza eseguendo il comando AWS CLI `register` dalla workstation o dall'istanza. Il modo più semplice per gestire l'operazione è utilizzando la registrazione guidata della console AWS OpsWorks Stacks, che semplifica il processo di costruzione della stringa di comando. Dopo aver preso familiarità con la procedura di registrazione, è possibile saltare la procedura guidata se si preferisce ed eseguire il comando `register`.

La sezione seguente descrive come utilizzare la procedura guidata per registrare l'istanza con uno stack esistente.

> **Note**
>
> Per registrare un'istanza con un nuovo stack, puoi farlo scegliendo Registra istanze nella dashboard Stacks. AWS OpsWorks Questo avvia una procedura guidata identica a quella degli stack esistenti, salvo per una pagina aggiuntiva che consente di configurare il nuovo stack.

Per utilizzare la procedura guidata di registrazione per registrare un'istanza

1. Nella console AWS OpsWorks Stacks, creare uno stack o aprire uno stack esistente.
2. Selezionare Instances (Istanze) nel riquadro di navigazione e selezionare register an instance (registra un'istanza).
3. Nella pagina Scegli un tipo di istanza, specifica se desideri registrare un'istanza Amazon EC2 o un'istanza locale:
   - Se stai registrando un' EC2 istanza Amazon, scegli Avanti: Seleziona istanze.
   - Se stai registrando un'istanza locale, scegli Avanti: Installa AWS CLI, quindi vai al passaggio 5.
4. Se stai registrando un' EC2 istanza Amazon, apri la pagina Seleziona istanze per selezionare l'istanza da registrare. AWS OpsWorks Stacks raccoglie le informazioni necessarie per creare il comando. Al termine, selezionare Next:Install AWS CLI (Successivo: installa interfaccia a riga di comando AWS).
5. L'istanza su cui intendi eseguire `register` deve eseguire la versione 1.16.180 di o una versione successiva. AWS CLI Per installare o aggiornare l' AWS CLI, la pagina dell'installazione guidata fornisce i collegamenti per le istruzioni di installazione e configurazione. Dopo aver verificato l'installazione dell' AWS CLI , specificare se il comando viene eseguito dall'istanza da registrare o da una workstation separata e quindi scegliere Next: Register Instances (Successivo: registra istanze).
6. La pagina Register Instances (Registra istanze) visualizza un modello per una stringa di comando `register` che incorpora le opzioni selezionate. Ad esempio, se stai registrando un' EC2 istanza Amazon da una workstation separata, il modello predefinito è simile al seguente.

```
aws opsworks register --infrastructure-class ec2 --region us-west-2
```

```
  --stack-id 247be7ea-3551-4177-9524-1ff804f453e3 --ssh-username [username] i-
f1245d10
```

> **Important**
>
> L'utente IAM creato durante il processo di registrazione è necessario per tutta la durata di un'istanza registrata. L'eliminazione dell'utente impedisce all'agente AWS OpsWorks Stacks di comunicare con il servizio. Per evitare problemi nella gestione delle istanze registrate nel caso in cui l'utente venga eliminato accidentalmente, aggiungi il `--use-instance-profile` parametro al `register` comando per utilizzare invece il profilo di istanza integrato dell'istanza. L'aggiunta del `--use-instance-profile` parametro impedisce inoltre che si verifichino errori quando si ruotano le chiavi di accesso AWS dell'account ogni 90 giorni (una best practice consigliata), poiché previene le discrepanze tra le chiavi di accesso disponibili per l' AWS OpsWorks agente e l'utente IAM richiesto.

Se imposti I use SSH keys su Yes, AWS OpsWorks Stacks aggiunge l'`--ssh-private-key`argomento alla stringa, che puoi usare per specificare un file di chiave SSH privato.

> **Note**
>
> Se desideri **`register`** accedere con una password, imposta I use SSH keys su No. Quando corri`register`, ti viene richiesta la password.

Copiare questa stringa in un editor di testo e modificarla in base alle esigenze. Tenere presente quanto indicato di seguito.

- Il testo tra parentesi rappresenta informazioni che è necessario fornire, come la posizione del file di chiave SSH.
- Il modello prevede che si stia eseguendo `register` con credenziali AWS predefinite. In caso contrario, aggiungere un argomento `--profile` alla stringa di comando e specificare il nome del profilo delle credenziali che si desidera utilizzare.

Per altri scenari, potrebbe essere necessario modificare il comando ulteriormente. Per una spiegazione degli argomenti `register` disponibili e metodi alternativi per costruire la stringa di comando, consulta Utilizzo del comando `register`. È anche possibile visualizzare la

documentazione del comando tramite l'esecuzione di `aws opsworks help register` dalla riga di comando. Per alcuni esempi di stringhe di comando, consulta Comandi register di esempio.

7. Dopo aver modificato la stringa di comando, aprire una finestra di terminale nella workstation o utilizzare SSH per accedere all'istanza ed eseguire il comando. L'operazione intera richiede in genere circa cinque minuti, durante i quali l'istanza si trova nello stato Registering (Registrazione).
8. Al termine dell'operazione, selezionare Done (Fatto). L'istanza è ora nello stato Registered (Registrato) ed è elencata come un'istanza non assegnata nella pagina Instances (Istanze) dello stack.

Il comando `register` funziona come indicato di seguito.

1. Se il comando `register` è in esecuzione in una workstation, utilizza prima SSH per l'accesso all'istanza da registrare.

   Il resto del processo viene eseguito sull'istanza e rimane lo stesso indipendentemente dal punto in cui è stato eseguito il comando.
2. Scarica il pacchetto dell'agente AWS OpsWorks Stacks da Amazon S3.
3. Decomprime e installa l'agente e le relative dipendenze, ad esempio AWS SDK per Ruby.
4. Crea quanto segue
   - Un utente IAM che avvia l'agente con il servizio AWS OpsWorks Stacks per fornire comunicazioni sicure.

     Le autorizzazioni dell'utente consentono solo l'operazione `opsworks:RegisterInstance` e scadono dopo 15 minuti.
   - Un gruppo IAM per lo stack, che contiene gli utenti delle istanze registrate.
5. Crea una coppia di chiavi RSA e invia la chiave pubblica a AWS OpsWorks Stacks.

   Questa coppia di chiavi viene utilizzata per crittografare le comunicazioni tra l'agente e AWS OpsWorks Stacks.
6. Registra l'istanza con Stacks. AWS OpsWorks Lo stack esegue un set di ricette di configurazione iniziale per configurare l'istanza, che comprende quanto segue.
   - Sovrascrittura del file hosts dell'istanza.

Registrando l'istanza, hai affidato la gestione degli utenti a AWS OpsWorks Stacks, che deve disporre del proprio file hosts per controllare le autorizzazioni di accesso SSH.

- Per EC2 le istanze Amazon, la configurazione iniziale include anche la registrazione di tutti i volumi Amazon EBS o gli indirizzi IP elastici collegati nello stack.

  Devi assicurarti che i volumi Amazon EBS non siano montati su punti di montaggio riservati, inclusi `/var/www` eventuali punti di montaggio riservati dai livelli dell'istanza. Per ulteriori informazioni sulla gestione delle risorse stack, consulta Gestione delle risorse. Per ulteriori informazioni sui punti di montaggio di livello, consulta AWS OpsWorks Stacks Layer Reference.

Per una descrizione completa delle modifiche della configurazione iniziale, consulta Configurazione iniziale delle modifiche di configurazione.

> **Note**
>
> La configurazione iniziale non aggiorna il sistema operativo di un'istanza registrata, è necessario gestire tale l'attività da se stessi. Per ulteriori informazioni, consulta Gestione degli aggiornamenti della sicurezza.

Utilizzo del comando **`register`**

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questa funzionalità è supportata solo per stack Linux.

Per registrare un'istanza, verifica che sia in esecuzione almeno la versione 1.16.180 dell' AWS CLI. Segue la sintassi generale del comando `register`.

```
aws opsworks register \
  [--profile profile_name] \
  [--region region_name] \
  --infrastructure-class instance_type \
  --stack-id stack ID \
  [--local] | [--ssh-private-key key_file --ssh-username username] | [--override-
ssh command_string] \
  [--override-hostname hostname] \
  [--debug] \
  [--override-public-ip public IP] \
  [--override-private-ip private IP] \
..[--use-instance-profile] \
  [ [IP address] | [hostname] | [instance ID]
```

I seguenti argomenti sono comuni a tutti AWS CLI i comandi.

`--profile`

(Facoltativo) Il nome del profilo delle credenziali. Se si omette questo argomento, il comando viene eseguito con le credenziali predefinite. Per ulteriori informazioni su come AWS CLI determina le credenziali predefinite, consulta Configurazione dell'interfaccia a riga di comando AWS.

`--region`

(Facoltativo) La regione dell'endpoint del servizio AWS OpsWorks Stacks. Non impostare la `--region` regione dello stack. AWS OpsWorks Stacks determina automaticamente la regione dello stack dall'ID dello stack.

> **Note**
>
> Se la tua regione predefinita è già impostata, puoi omettere questo argomento. Per ulteriori informazioni su come specificare una regione predefinita, consulta Configurazione dell'interfaccia a riga di comando AWS.

Utilizza i seguenti argomenti sia per Amazon EC2 che per le istanze locali.

`--infrastructure-class`

(Obbligatorio) Questo parametro deve essere impostato su uno `ec2` o `on-premises`, rispettivamente, per indicare se stai registrando un'istanza Amazon EC2 o locale.

`--stack-id`

(Obbligatorio) L'ID dello stack con il quale l'istanza deve essere registrata.

> **Note**
>
> Per trovare un ID dello stack, nella pagina Stack, scegliere Settings (Impostazioni). L'ID dello stack è denominato OpsWorks ID ed è un GUID simile. `ad21bce6-7623-47f1-bf9d-af2affad8907`

Argomenti accesso SSH

Utilizzare i seguenti argomenti per specificare il modo in cui `register` deve effettuare l'accesso all'istanza.

`--local`

(Facoltativo) Utilizzare questo argomento per registrare l'istanza sulla quale si esegue il comando.

In questo caso, `register` non deve accedere all'istanza.

`--ssh-private-key` e **`--ssh-username`**

(Facoltativo) Utilizzare questi argomenti se si sta registrando l'istanza da una workstation separata e si desidera specificare esplicitamente il nome utente o il file di chiave privata.

- `--ssh-username`— Utilizzate questo argomento per specificare un nome utente SSH.

  Se si omette `--ssh-username`, `ssh` usa il nome utente predefinito.

- `--ssh-private-key`— Utilizzate questo argomento per specificare in modo esplicito un file di chiave privata.

  Se si omette `--ssh-private-key`, `ssh` tenterà di effettuare l'accesso utilizzando tecniche di autenticazione che non richiedono una password, incluso l'utilizzo di una chiave privata predefinita. Se nessuna di tali tecniche è supportata, `ssh` chiede la password. Per ulteriori

informazioni su come `ssh` gestisce l'autenticazione, consulta Il protocollo di autenticazione Secure Shell (SSH).

`--override-ssh`

(Facoltativo) Utilizzare questo argomento se si sta registrando l'istanza da un'altra workstation e si desidera specificare una stringa di comando personalizzata `ssh`. Il comando `register` usa questa stringa di comando per accedere all'istanza registrata.

Per ulteriori informazioni sulla funzionalità `ssh`, consultare SSH.

`--override-hostname`

(Facoltativo) Specifica un nome host per l'istanza, che viene utilizzato solo da AWS OpsWorks Stacks. Il valore predefinito è il nome host dell'istanza.

`--debug`

(Facoltativo), fornisce le informazioni di debug nel caso in cui il processo di registrazione abbia esito negativo. Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di registrazione delle istanze.

`--use-instance-profile`

(Facoltativo, ma altamente consigliato per EC2 le istanze Amazon) Consente al `register` comando di utilizzare un profilo di istanza collegato, anziché creare un utente IAM. L'aggiunta di questo parametro può aiutare a prevenire gli errori che si verificano se si tenta di gestire un'istanza registrata quando l'utente IAM è stato eliminato accidentalmente.

> **Important**
>
> L'utente IAM creato durante il processo di registrazione è necessario per tutta la durata di un'istanza registrata. L'eliminazione dell'utente impedisce all'agente AWS OpsWorks Stacks di comunicare con il servizio. Per evitare problemi nella gestione delle istanze registrate nel caso in cui l'utente venga eliminato accidentalmente, aggiungi il `--use-instance-profile` parametro al `register` comando per utilizzare invece il profilo di istanza integrato dell'istanza. L'aggiunta del `--use-instance-profile` parametro evita inoltre che si verifichino errori quando si ruotano le chiavi di accesso AWS dell'account ogni 90 giorni (una procedura consigliata), poiché previene le discrepanze tra le chiavi di accesso disponibili per l'agente e l' AWS OpsWorks utente richiesto.

Target

(Condizionale) Se si esegue il comando da una workstation, il valore finale nella stringa di comando specifica il target di registrazione in uno dei seguenti modi.

- L'indirizzo IP pubblico dell'istanza.
- Il nome host dell'istanza.
- Per le EC2 istanze Amazon, l'ID dell'istanza.

AWS OpsWorks Stacks utilizza l'ID dell'istanza per ottenere la configurazione dell'istanza, incluso l'indirizzo IP pubblico dell'istanza. Per impostazione predefinita, AWS OpsWorks Stacks utilizza questo indirizzo per costruire la stringa di `ssh` comando che utilizza per accedere all'istanza. Se si ha bisogno di connettersi a un indirizzo IP privato, è necessario utilizzare `--override-ssh` per fornire una stringa di comando personalizzata. Per vedere un esempio, consulta Registrare un'istanza locale da una workstation.

> **Note**
>
> Se viene specificato un nome host, `ssh` dipende dal server DNS per risolvere il nome di una determinata istanza. Se non si è certi che il nome host sia univoco, utilizzare `ssh` per verificare che il nome host si risolva nella corretta istanza.

Se si esegue il comando dall'istanza da registrare, omettere l'identificatore di istanza e utilizzare invece l'argomento `--local`.

I seguenti argomenti sono solo per istanze locali.

`--override-public-ip`

(Facoltativo) AWS OpsWorks Stacks visualizza l'indirizzo specificato come indirizzo IP pubblico dell'istanza. Non modifica l'indirizzo IP pubblico dell'istanza. Tuttavia, se un utente utilizza la console per connettersi all'istanza, ad esempio scegliendo l'indirizzo nella pagina Istanze, AWS OpsWorks Stacks utilizza l'indirizzo specificato. AWS OpsWorks Stacks determina automaticamente il valore predefinito dell'argomento.

`--override-private-ip`

(Facoltativo) AWS OpsWorks Stacks visualizza l'indirizzo specificato come indirizzo IP privato dell'istanza. Non modifica l'indirizzo IP privato dell'istanza. AWS OpsWorks Stacks determina automaticamente il valore predefinito dell'argomento.

Comandi register di esempio

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questa funzionalità è supportata solo per stack Linux.

Questa sezione contiene alcuni esempi di stringhe di comando `register`.

Registra un' EC2 istanza Amazon da una workstation

L'esempio seguente registra un' EC2 istanza Amazon da una workstation. La stringa di comando utilizza credenziali predefinite e identifica l'istanza tramite il relativo ID di EC2 istanza Amazon. È possibile utilizzare l'esempio per istanze locali modificando `ec2` con `on-premises`.

```
aws opsworks register \
  --region us-west-2 \
  --use-instance-profile \
  --infrastructure-class ec2 \
  --stack-id ad21bce6-7623-47f1-bf9d-af2affad8907 \
  --ssh-user-name my-sshusername \
  --ssh-private-key "./keys/mykeys.pem" \
  i-2422b9c5
```

Registrare un'istanza locale da una workstation

L'esempio seguente registra un'istanza locale da un'altra workstation. La stringa di comando utilizza le credenziali predefinite ed effettua l'accesso all'istanza con la stringa di comando specificata `ssh`. Se l'istanza richiede una password, `register` la chiede. Puoi utilizzare l'esempio per le EC2 istanze Amazon cambiando `on-premises` in`ec2`.

```
aws opsworks register \
  --region us-west-2 \
  --infrastructure-class on-premises \
  --stack-id ad21bce6-7623-47f1-bf9d-af2affad8907 \
  --override-ssh "ssh your-user@192.0.2.0"
```

> **Note**
>
> Puoi usarlo `--override-ssh` per specificare qualsiasi stringa di comando SSH personalizzata. AWS OpsWorks Stacks utilizza quindi la stringa specificata per accedere all'istanza invece di creare una stringa di comando. Per un altro esempio, consulta Registrare un'istanza utilizzando una stringa di comando SSH personalizzata.

Registrare un'istanza utilizzando una stringa di comando SSH personalizzata

L'esempio seguente registra un'istanza locale da una workstation e utilizza l'`--override-ssh`argomento per specificare un comando SSH personalizzato da `register` utilizzare per accedere all'istanza. Questo esempio utilizza `sshpass` per accedere con un nome utente e una password, ma è possibile specificare una qualsiasi stringa di comando `ssh` valida.

```
aws opsworks register \
  --region us-west-2 \
  --infrastructure-class on-premises \
  --stack-id 2f92ff9d-04f2-4728-879b-f4283b40783c \
  --override-ssh "sshpass -p 'mypassword' ssh your-user@192.0.2.0"
```

Registrare un'istanza eseguendo **`register`** dall'istanza

L'esempio seguente mostra come registrare un' EC2 istanza Amazon eseguendola `register` dall'istanza stessa. La stringa di comando varia a seconda delle credenziali predefinite per le autorizzazioni. Per utilizzare l'esempio per un'istanza locale, passa `--infrastructure-class` a`on-premises`.

```
aws opsworks register \
  --region us-west-2 \
  --infrastructure-class ec2 \
  --stack-id ad21bce6-7623-47f1-bf9d-af2affad8907 \
  --local
```

Registrare un'istanza con un indirizzo IP privato

Come impostazione predefinita, `register` utilizza l'indirizzo IP pubblico dell'istanza per accedere all'istanza. Per registrare un'istanza con un indirizzo IP privato, ad esempio un'istanza in una sottorete privata di un VPC, è necessario utilizzare `--override-ssh` per specificare una stringa di comando `ssh` personalizzata.

```
aws opsworks register \
  --region us-west-2 \
  --infrastructure-class ec2 \
  --stack-id 2f92ff9d-04f2-4728-879b-f4283b40783c \
  --override-ssh "ssh -i mykey.pem ec2-user@10.183.201.93" \
  i-2422b9c5
```

Policy di registrazione delle istanze

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le `AWSOpsWorksRegisterCLI_OnPremises` politiche `AWSOpsWorksRegisterCLI_EC2` e forniscono le autorizzazioni corrette rispettivamente per la registrazione delle istanze EC2 e quelle locali. Aggiungi `AWSOpsWorksRegisterCLI_EC2` al tuo utente IAM per registrare EC2 le istanze, ma aggiungi al tuo utente `AWSOpsWorksRegisterCLI_OnPremises` per registrare le istanze locali. Per utilizzare queste policy, devi eseguire almeno la versione 1.16.180 o una versione successiva. AWS CLI

La policy **AWSOpsWorksRegisterCLI_EC2**

Aggiungilo `AWSOpsWorksRegisterCLI_EC2` al tuo utente per registrare le istanze. EC2 È necessario utilizzare questo profilo se si prevede di registrare solo EC2 istanze. Quando utilizzi questa politica, le autorizzazioni vengono fornite dal profilo dell' EC2istanza dell'istanza.

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "opsworks:AssignInstance",
            "opsworks:CreateLayer",
            "opsworks:DeregisterInstance",
            "opsworks:DescribeInstances",
            "opsworks:DescribeStackProvisioningParameters",
            "opsworks:DescribeStacks",
            "opsworks:UnassignInstance"
          ],
          "Resource": [
            "*"
          ]
        },
        {
          "Effect": "Allow",
          "Action": [
            "ec2:DescribeInstances"
          ],
          "Resource": [
            "*"
          ]
        }
      ]
    }
```

La policy **AWSOpsWorksRegisterCLI_OnPremises**

Aggiungilo `AWSOpsWorksRegisterCLI_OnPremises` al tuo utente per registrare le istanze locali. Questa politica include le autorizzazioni IAM, ad esempio`AttachUserPolicy`, ma le risorse su cui funzionano tali autorizzazioni sono limitate.

```
    {
```

```
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "opsworks:AssignInstance",
          "opsworks:CreateLayer",
          "opsworks:DeregisterInstance",
          "opsworks:DescribeInstances",
          "opsworks:DescribeStackProvisioningParameters",
          "opsworks:DescribeStacks",
          "opsworks:UnassignInstance"
        ],
        "Resource": [
          "*"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "ec2:DescribeInstances"
        ],
        "Resource": [
          "*"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "iam:CreateGroup",
          "iam:AddUserToGroup"
        ],
        "Resource": [
          "arn:aws:iam::*:group/AWS/OpsWorks/OpsWorks-*"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "iam:CreateUser",
          "iam:CreateAccessKey"
        ],
        "Resource": [
          "arn:aws:iam::*:user/AWS/OpsWorks/OpsWorks-*"
```

```
          ]
        },
        {
          "Effect": "Allow",
          "Action": [
            "iam:AttachUserPolicy"
          ],
          "Resource": [
            "arn:aws:iam::*:user/AWS/OpsWorks/OpsWorks-*"
          ],
          "Condition": {
            "ArnEquals":
              {
                "iam:PolicyARN": "arn:aws:iam::aws:policy/
AWSOpsWorksInstanceRegistration"
              }
            }
        }
      ]
    }
```

(Obsoleta) La policy **`AWSOpsWorksRegisterCLI`**

> **Important**
>
> La policy `AWSOpsWorksRegisterCLI` è obsoleta e non può essere utilizzata per registrare le nuove istanze. È disponibile solo per la compatibilità con le versioni precedenti delle istanze che sono già state registrate. La `AWSOpsWorksRegisterCLI` policy include molte autorizzazioni IAM`CreateUser`, tra cui, e`PutUserPolicy`. `AddUserToGroup` Poiché queste sono autorizzazioni a livello di amministratore, devi assegnare la policy `AWSOpsWorksRegisterCLI` solo agli utenti amministratori affidabili.

Gestione di istanze registrate

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questa funzionalità è supportata solo per stack Linux.

Quando registri un'istanza, questa diventa un'istanza AWS OpsWorks Stacks e puoi gestirla più o meno allo stesso modo delle istanze create con Stacks. AWS OpsWorks Sussistono due principali differenze:

- Le istanze registrate non devono essere assegnate a un livello.
- È possibile annullare la registrazione di un'istanza registrata e restituirla al proprio controllo diretto.

Dopo aver registrato un'istanza, questa si trova nello stato Registrato. AWS OpsWorks Stacks offre le seguenti funzionalità di gestione a tutte le istanze registrate:

- Health checks: AWS OpsWorks Stacks monitora l'agente per valutare se l'istanza continua a funzionare.

  Se un'istanza non supera un controllo di integrità, AWS OpsWorks Stacks corregge automaticamente le istanze Amazon registrate e modifica lo stato delle EC2 istanze locali registrate in. `connection lost`
- CloudWatch monitoraggio: il monitoraggio è abilitato per le istanze CloudWatch registrate.

  È possibile monitorare i parametri, ad esempio l'utilizzo della CPU e la memoria disponibile e, facoltativamente, ricevere una notifica nel caso in cui un parametro superi una soglia specificata.
- Gestione degli utenti: AWS OpsWorks Stacks offre un modo semplice per specificare quali utenti possono accedere all'istanza e quali operazioni sono autorizzati a eseguire. Per ulteriori informazioni, consulta Gestione delle autorizzazioni utente.
- Esecuzione delle ricette: è possibile utilizzare il comando Execute Recipes stack per eseguire le ricette di Chef sull'istanza.
- Aggiornamenti del sistema operativo: puoi utilizzare il comando Update Dependencies stack per aggiornare il sistema operativo dell'istanza.

Per sfruttare appieno la funzionalità di gestione degli AWS OpsWorks stack, puoi assegnare l'istanza a un livello. Per ulteriori informazioni, consulta Assegnazione di un'istanza registrata a un livello.

Esistono differenze tra il modo in cui AWS OpsWorks Stacks gestisce Amazon EC2 e le istanze locali.

EC2 Istanze Amazon

- Se interrompi un' EC2 istanza Amazon registrata, AWS OpsWorks Stacks interrompe le istanze supportate dall'archivio delle istanze e interrompe le istanze supportate da Amazon EBS.

  L'istanza è ancora registrata con lo stack ed è assegnata ai suoi livelli, così è possibile riavviarla se necessario. È necessario annullare la registrazione di un'istanza registrata per rimuoverla da uno stack, esplicitamente o eliminando l'istanza, che automaticamente annulla la registrazione.
- Se riavvii un' EC2 istanza Amazon registrata o l'istanza fallisce e viene riparata automaticamente, il risultato è lo stesso dell'arresto e del riavvio dell'istanza utilizzando Amazon. EC2 Notare queste differenze:
  - Istanze supportate dall'archivio di istanze: AWS OpsWorks Stacks avvia una nuova istanza con la stessa AMI.

    Tieni presente che AWS OpsWorks Stacks non è a conoscenza di alcuna operazione eseguita sull'istanza prima della registrazione, come l'installazione di pacchetti software. Se desideri che AWS OpsWorks Stacks installi pacchetti o esegua altre attività di configurazione all'avvio, devi fornire ricette Chef personalizzate che eseguano le attività richieste e assegnarle agli eventi di configurazione dei livelli appropriati.
  - Istanze supportate da Amazon EBS: AWS OpsWorks Stacks avvia una nuova istanza con la stessa AMI e ricollega il volume root, che ripristina l'istanza alla configurazione precedente.
- Se annulli la registrazione di un' EC2 istanza Amazon registrata, questa torna a essere un'istanza Amazon EC2 normale.

Istanze locali

- AWS OpsWorks Gli stacks non possono arrestare o avviare un'istanza locale registrata.

  L'annullamento dell'assegnazione di un'istanza locale registrata attiva un evento Shutdown. Tuttavia, quell'evento esegue semplicemente le ricette Shutdown dei livelli assegnati. Eseguono attività come spegnere i servizi, ma non arrestano l'istanza.
- AWS OpsWorks Gli stacks non possono riparare automaticamente un'istanza locale registrata in caso di errore, ma l'istanza viene contrassegnata come connessione persa.

- Le istanze locali non possono utilizzare i servizi di indirizzi Elastic Load Balancing, Amazon EBS o Elastic IP.

Assegnazione di un'istanza registrata a un livello

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questa funzionalità è supportata solo per stack Linux.

Dopo aver registrato un'istanza, è possibile assegnarla a uno o più livelli. Il vantaggio di assegnare un'istanza a un layer anziché lasciarla non assegnata è che è possibile assegnare ricette personalizzate agli eventi del ciclo di vita del layer. AWS OpsWorks Stacks li esegue quindi automaticamente al momento opportuno, dopo le ricette del layer per quell'evento.

- È possibile assegnare qualsiasi istanza registrata a un livello personalizzato. Un livello personalizzato ha un set minimo di ricette che non installano alcun pacchetto, quindi non devono creare conflitto con la configurazione esistente dell'istanza.
- È possibile assegnare istanze locali ai AWS OpsWorks livelli integrati di Stacks.

  Ogni livello integrato include ricette che installano automaticamente uno o più pacchetti. Ad esempio, le ricette di Java App Server Setup installano Apache e Tomcat. Le ricette del livello possono anche eseguire altre operazioni, ad esempio il riavvio di servizi e la distribuzione di applicazioni. Prima di assegnare un'istanza locale a un livello integrato, è necessario assicurarsi che le ricette del layer non creino conflitti, ad esempio il tentativo di installare una versione dell'application server diversa da quella attualmente presente nell'istanza. Per ulteriori informazioni, consulta Livelli e AWS OpsWorks Stacks Layer Reference.

Per assegnare un'istanza registrata a un livello

1. Aggiungere i livelli che si desidera utilizzare allo stack, se non è già stato fatto.
2. Scegliere Instances (Istanze) nel riquadro di navigazione e scegliere assign (assegna) nella colonna Actions (Operazioni) dell'istanza.
3. Selezionare i livelli appropriati e scegliere Save (Salva).

Quando si assegna un'istanza a un layer, AWS OpsWorks Stacks esegue le seguenti operazioni.

- Esegue le ricette di impostazione del livello
- Aggiunge eventuali indirizzi IP elastici o volumi Amazon EBS collegati alle risorse dello stack.

  Puoi quindi utilizzare AWS OpsWorks Stacks per gestire queste risorse. Per ulteriori informazioni, consulta Gestione delle risorse.

Al termine, l'istanza passa allo stato online ed è completamente incorporata nello stack. AWS OpsWorks Stacks esegue quindi le ricette assegnate al layer ogni volta che si verifica un evento del ciclo di vita.

Annullamento dell'assegnazione di un'istanza registrata

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Note

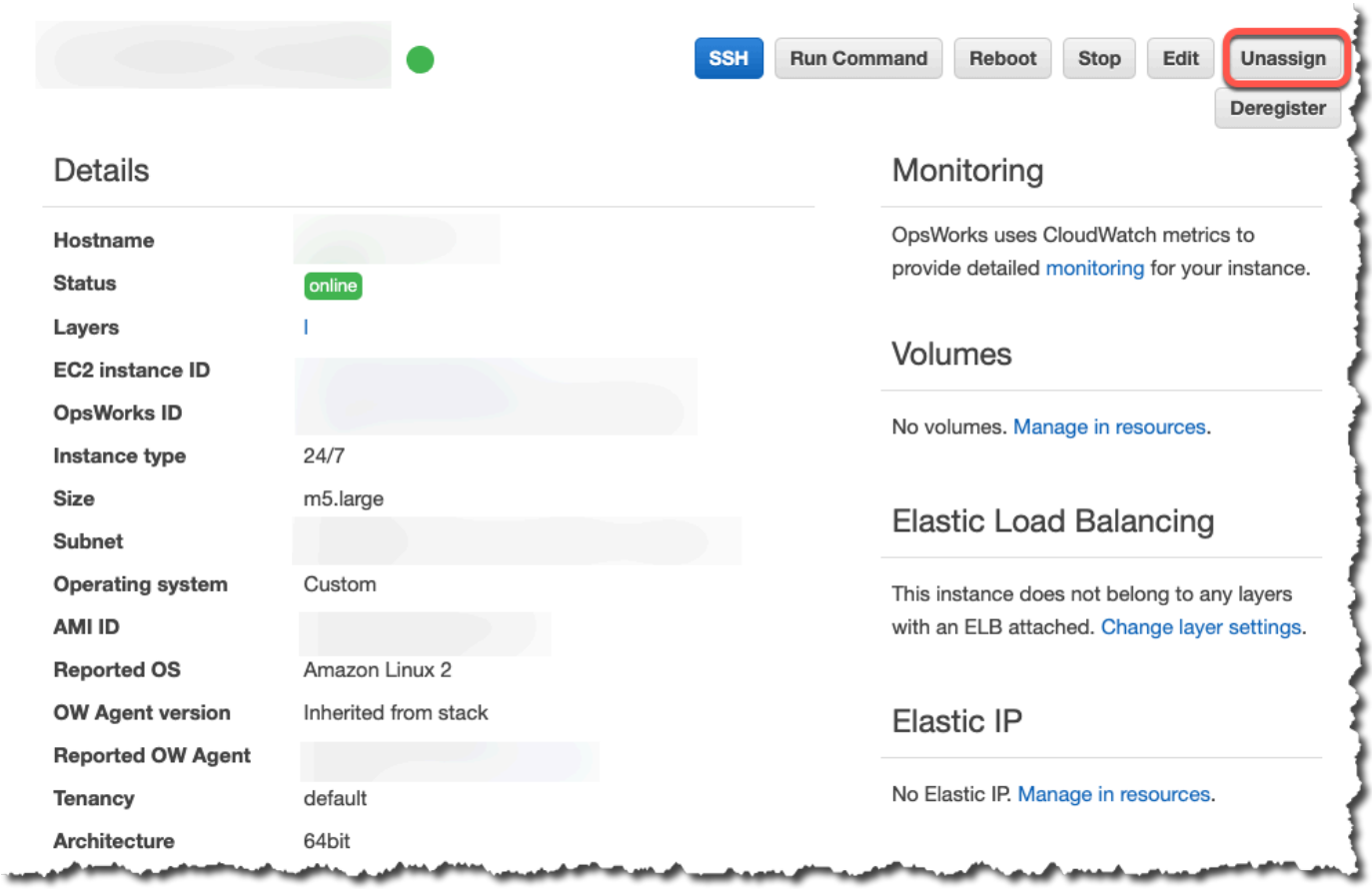
Questa funzionalità è supportata solo per stack Linux.

Puoi annullare l'assegnazione di un'istanza registrata dai suoi livelli utilizzando la AWS OpsWorks console o il funzionamento dell' AWS CLI SDK.

Quando annulli l'assegnazione di un'istanza, AWS OpsWorks Stacks esegue le ricette di spegnimento del layer sull'istanza. Queste ricette eseguono attività come spegnere i servizi, ma non arrestano l'istanza. Se l'istanza è assegnata a più livelli, l'annullamento dell'assegnazione si applica a ogni livello; non è possibile annullare l'assegnazione di un'istanza da un sottoinsieme di livelli. Tuttavia, l'istanza è ancora registrata con lo stack ed è possibile assegnarla a un altro livello, se desiderato.

Per annullare l'assegnazione di un'istanza registrata utilizzando la console

1. Nel pannello di navigazione, seleziona Instances (Istanze).
2. Scegli l'istanza che desideri annullare l'assegnazione.
3. Nella pagina Dettagli dell'istanza, scegli Annulla assegnazione.



Per annullare l'assegnazione di un'istanza registrata utilizzando il AWS CLI

Eseguite il aws opsworks unassign-instancecomando per annullare l'assegnazione di un'istanza registrata a tutti i livelli che utilizzano l'istanza.

```
aws opsworks unassign-instance --region region --instance-id instance-id
```

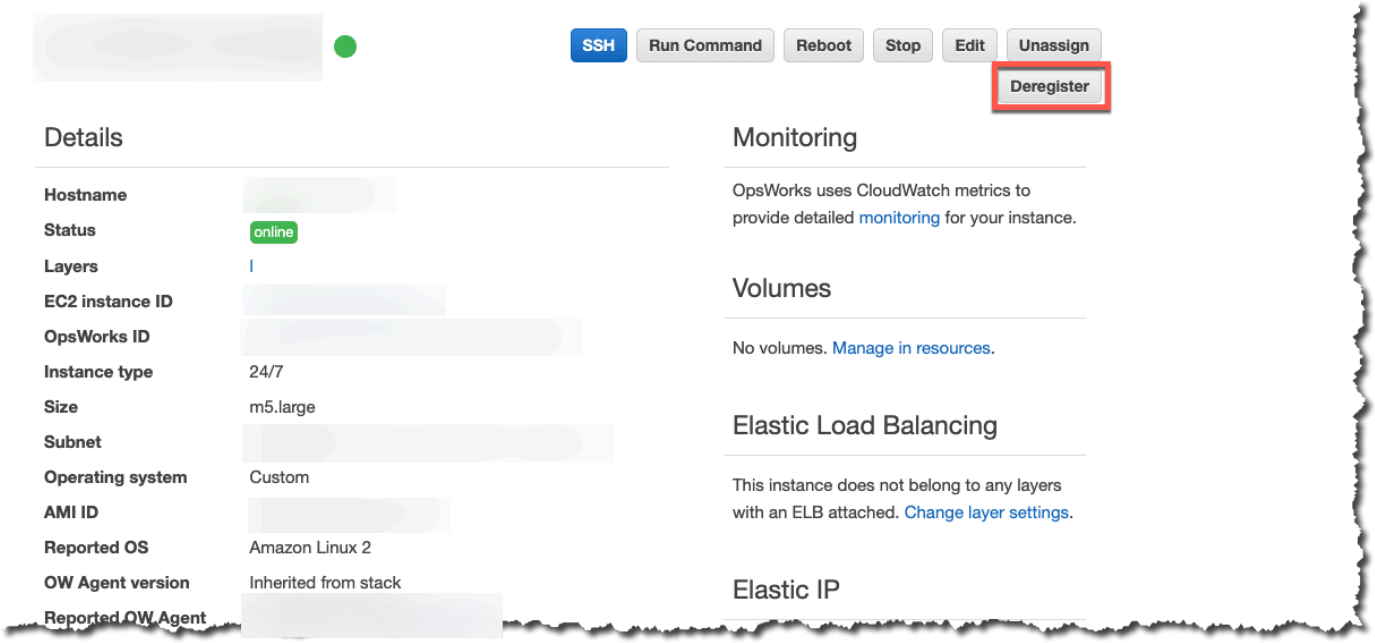
Annullamento della registrazione di un'istanza registrata

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi annullare la registrazione di un'istanza utilizzando la AWS OpsWorks console o il funzionamento dell' AWS CLI SDK.

Per annullare la registrazione di un'istanza utilizzando la console

1. Nel pannello di navigazione, seleziona Instances (Istanze).
2. Scegli l'istanza che desideri annullare la registrazione.
3. Nella pagina Dettagli dell'istanza, scegli Annulla registrazione.



Per annullare la registrazione di un'istanza utilizzando il AWS CLI

Eseguite il aws opsworks deregister-instancecomando per annullare la registrazione di un'istanza dal relativo stack.

```
aws opsworks deregister-instance --region region --instance-id instance-id
```

Quando annulli la registrazione di un'istanza, AWS OpsWorks Stacks esegue le seguenti operazioni:

- Rimuove l'istanza dallo stack.
- Annulla l'assegnazione dell'istanza da qualsiasi livello assegnato.
- Spegne e disinstalla l'agente.
- Annulla la registrazione di tutte le risorse allegate (indirizzi IP elastici e volumi Amazon EBS).

  Questa procedura include le risorse che erano collegate all'istanza prima della registrazione e le risorse che hai utilizzato AWS OpsWorks Stacks per collegare all'istanza mentre faceva parte dello stack. Dopo l'annullamento della registrazione, le risorse non sono più parte delle risorse dello stack, ma rimangono collegate all'istanza.
- Per istanze locali, interrompe gli addebiti.
- Rimuove tutti i tag OpsWorks aggiunti all'istanza.

L'istanza rimane in esecuzione, ma è sotto il tuo controllo diretto e non è più gestita da AWS OpsWorks Stacks.

> **Note**
>
> Sia la registrazione che l'annullamento della registrazione dei computer o delle istanze sono completamente supportate solo negli stack Linux. Per gli stack Windows, l'annullamento della registrazione delle istanze è consentito, ma non comporta la disinstallazione dell'agente dall'istanza. OpsWorks L'annullamento della registrazione non rimuove tutti i file modificati e non ripristina completamente copie di backup di alcuni file. Questo elenco si riferisce a stack Chef 11.10 e Chef 12; le differenze tra le due versioni sono annotate in questa pagina.
>
> - `/etc/hosts` ha il backup in `/var/lib/aws/opsworks/local-mode-cache/backup/etc/`, ma non viene ripristinato.
> - Le voci rimangono per `aws` e `opsworks` in passwd, group, file shadow, ecc.
> - `/etc/sudoers`contiene un riferimento a una directory Stacks. AWS OpsWorks

- I seguenti file possono essere ignorati; a lungo termine, considerare l'eliminazione di `/var/lib/aws/opsworks`.
  - `/var/log/aws/opsworks` rimane su istanze in stack Chef 11.10.
  - `/var/lib/aws/opsworks` rimane su stack Chef 11.10 e Chef 12.
  - `/var/chef` rimane su istanze in stack Chef 12.
- Altri file ignorati:
  - `/etc/logrotate.d/opsworks-agent`
  - `/etc/cron.d/opsworks-agent-updater`
  - `/etc/ld.so.conf.d/opsworks-user-space.conf`
  - `/etc/motd.opsworks-static`
  - `/etc/aws/opsworks`
  - `/etc/sudoers.d/opsworks`
  - `/etc/sudoers.d/opsworks-agent`

Ciclo di vita dell'istanza registrata

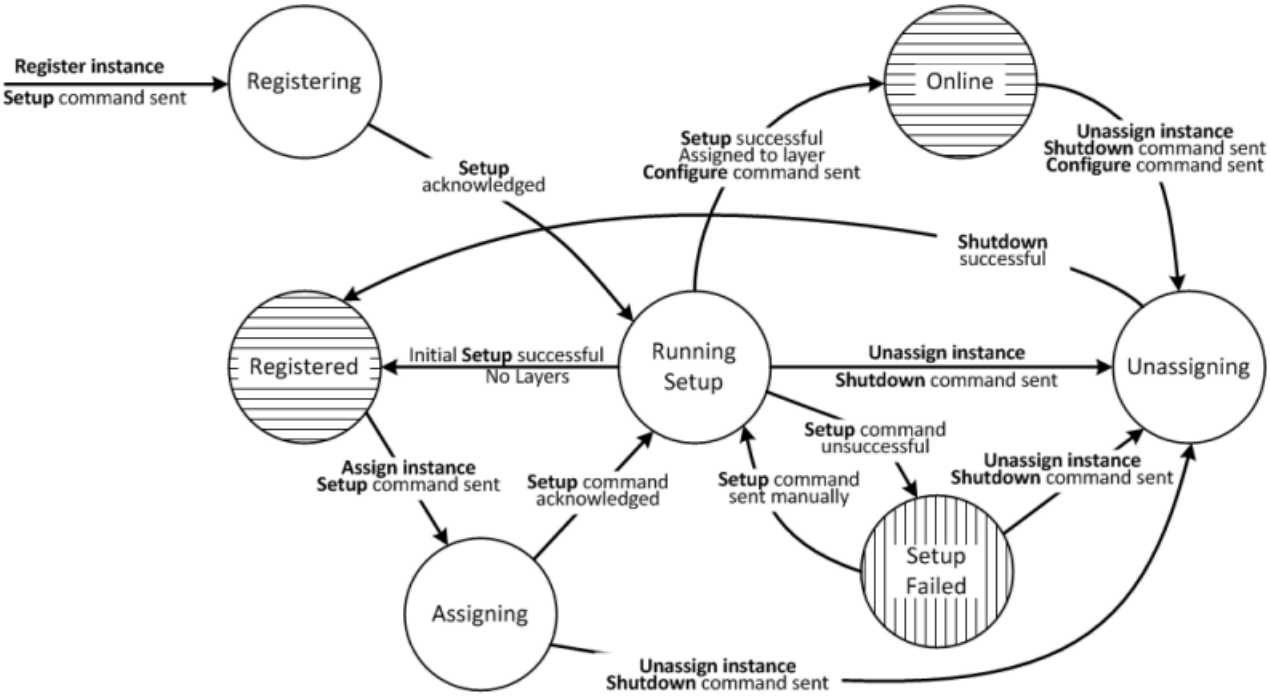
Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Note

Questa funzionalità è supportata solo per stack Linux.

Il ciclo di vita dell'istanza registrata inizia dopo che l'agente è stato installato ed è in esecuzione. A quel punto, ordina a AWS OpsWorks Stacks di registrare l'istanza nello stack. Il seguente diagramma di stato riepiloga gli elementi chiave del ciclo di vita.



Ogni stato corrisponde a uno stato di istanza. I bordi rappresentano uno dei seguenti AWS OpsWorks comandi Stacks. I dettagli vengono approfonditi nelle sezioni seguenti.

- Setup: questo comando corrisponde all'evento Setup lifecycle ed esegue le ricette di installazione dell'istanza.
- Configure: questo comando corrisponde all'evento Configure lifecycle.

  AWS OpsWorks Stacks attiva questo evento su ogni istanza dello stack quando un'istanza entra o esce dallo stato online. Le istanze eseguono le ricette di configurazione, che apportano le modifiche necessarie per supportare la nuova istanza.
- Shutdown: questo comando corrisponde all'evento del ciclo di vita Shutdown, che esegue le ricette Shutdown dell'istanza.

  Queste ricette eseguono attività come spegnere i servizi, ma non arrestano l'istanza.
- Annulla registrazione: questo comando annulla la registrazione di un'istanza e non corrisponde a un evento del ciclo di vita.

> **Note**
>
> Per semplicità il diagramma non mostra l'annullamento della registrazione e gli stati eliminati. È possibile annullare la registrazione di un'istanza da qualsiasi degli stati nel diagramma,

che invia un comando Annulla registrazione all'istanza e lo sposta nello stato Annullamento registrazione.

- Se annulli la registrazione di un'istanza online, AWS OpsWorks Stacks invia un comando Configure alle istanze rimanenti dello stack per notificare loro che l'istanza sta andando offline.
- Dopo che il comando Annulla registrazione viene confermato, l'istanza è ancora in esecuzione, ma è ancora nello stato Eliminato e non più parte dello stack. Se si desidera di nuovo incorporare l'istanza nel nuovo stack, è necessario registrarla di nuovo.

Argomenti

- Registrazione in corso
- Esecuzione impostazione
- Registered
- Assegnazione
- Online
- Errore configurazione
- Annullamento dell'assegnazione
- Configurazione iniziale delle modifiche di configurazione

Registrazione in corso

Dopo che l'agente ha inviato una richiesta di registrazione, AWS OpsWorks Stacks avvia il ciclo di vita dell'istanza inviando un comando di installazione all'istanza, mettendola nello stato di registrazione. Dopo che l'istanza riconosce il comando Imposta, passa allo stato Esecuzione impostazione.

Esecuzione impostazione

Lo stato Esecuzione configurazione esegue le ricette di impostazione dell'istanza. L'impostazione funziona a seconda dello stato precedente.

> **Note**
>
> Se si annulla l'assegnazione dell'istanza mentre si trova nello stato Running Setup, AWS OpsWorks Stacks invia un comando Shutdown, che esegue le ricette di chiusura dell'istanza ma non la arresta. L'istanza passa allo stato Annullamento dell'assegnazione.

Argomenti

- Registrazione in corso
- Assegnazione
- Errore configurazione

Registrazione in corso

Durante il processo di registrazione, il setup crea un'istanza AWS OpsWorks Stacks per rappresentare l'istanza registrata nello stack ed esegue una serie di ricette di installazione di base sull'istanza.

Una modifica chiave eseguita dalla configurazione iniziale corrisponde alla sovrascrittura del file hosts dell'istanza. Registrando l'istanza, viene passata la gestione dell'utente a AWS OpsWorks Stacks, che deve avere il proprio file hosts per controllare le autorizzazioni di accesso SSH. La configurazione iniziale, inoltre, crea o modifica un numero di file e, nei sistemi Ubuntu, modifica le origini del pacchetto e installa un set di pacchetti. Per informazioni dettagliate, consultare Configurazione iniziale delle modifiche di configurazione.

Durante la registrazione, il processo richiama l'IAM `AttachUserPolicy` che fa parte delle autorizzazioni associate all'utente IAM che crei come prerequisito. Se `AttachUserPolicy` non esiste, in genere per il fatto che si sta eseguendo una versione precedente dell'interfaccia a riga di comando di AWS), il processo esegue il fallback alla chiamata `PutUserPolicy`.

> **Note**
>
> Per motivi di coerenza, AWS OpsWorks Stacks esegue tutte le principali ricette di installazione. Tuttavia, alcuni di esse eseguono alcune o tutte le attività solo se un'istanza è stata assegnata almeno a un livello, in modo che non influenzino necessariamente la configurazione iniziale.

- Se la configurazione viene completata correttamente, l'istanza passa allo stato Registered.
- Se la configurazione non viene completata correttamente, l'istanza passa allo stato Errore configurazione.

Assegnazione

All'istanza è assegnato almeno un layer. AWS OpsWorks Stacks esegue le ricette di configurazione di ogni livello, incluse tutte le ricette personalizzate assegnate all'evento Setup dei livelli.

- Se la configurazione viene completata correttamente, l'istanza passa allo stato Online e AWS OpsWorks Stacks attiva un evento configurazione del ciclo di vita per ogni istanza nello stack per avvisarli della nuova istanza.
- Se la configurazione non viene completata correttamente, l'istanza passa allo stato Errore configurazione.

> **Note**
>
> Questo processo di configurazione esegue le ricette chiave una seconda volta. Tuttavia, le ricette Chef sono idempotenti, quindi non ripetono attività che sono già state eseguite.

Errore configurazione

Se un processo di configurazione per un'istanza nello stato Assegnazione ha esito negativo, è possibile riprovare utilizzando il comando configurazione stack per eseguire nuovamente le ricette di impostazione dell'istanza

- Se la configurazione viene completata correttamente, l'istanza assegnata passa allo stato Online e AWS OpsWorks Stacks attiva un evento configurazione del ciclo di vita per ogni istanza nello stack per avvisarli della nuova istanza.
- Se il tentativo di configurazione non viene completato correttamente, l'istanza passa di nuovo allo stato Errore configurazione.

Registered

Le istanze nello stato Registered fanno parte dello stack e sono gestite da AWS OpsWorks Stacks ma non sono assegnate a un layer. Possono rimanere in questo stato a tempo indeterminato.

Se si assegna l'istanza a uno o più livelli, AWS OpsWorks Stacks invia un comando di configurazione all'istanza e questa passa allo stato. Assegnazione

Assegnazione

Dopo che l'istanza riconosce il comando Imposta, passa allo stato Esecuzione impostazione.

Se annulli l'assegnazione dell'istanza mentre è nello stato Assegnazione, AWS OpsWorks Stacks interrompe il processo di configurazione e invia un comando Shutdown. L'istanza passa allo stato Annullamento dell'assegnazione.

Online

L'istanza è ora un membro di almeno un livello e viene trattata come un'istanza AWS OpsWorks Stacks normale. Può rimanere in questo stato a tempo indeterminato.

Se annulli l'assegnazione dell'istanza mentre è nello stato Online, AWS OpsWorks Stacks invia un comando Shutdown all'istanza e un comando Configure alle altre istanze dello stack. L'istanza passa allo stato Annullamento dell'assegnazione.

Errore configurazione

Il comando di configurazione non è andato a buon fine.

- È possibile riprovare eseguendo il comando Imposta stack.

  L'istanza torna allo stato Esecuzione impostazione.
- Se annulli l'assegnazione dell'istanza, Stacks invia un comando Shutdown all'istanza. AWS OpsWorks

  L'istanza passa allo stato Annullamento dell'assegnazione.

Annullamento dell'assegnazione

Al termine del comando di arresto, l'istanza non è più assegnata ad alcun livello e torna allo stato Registered.

> **Note**
>
> Se un'istanza è assegnata a più livelli, l'annullamento dell'assegnazione si applica a ogni livello; non è possibile annullare l'assegnazione di un sottoinsieme di livelli assegnati. Se si

desidera un set diverso livelli assegnati, annullare l'assegnazione dell'istanza e riassegnare i livelli desiderati.

Configurazione iniziale delle modifiche di configurazione

La configurazione iniziale crea o modifica i seguenti file e directory su tutte le istanze registrate.

File creati

```
/etc/apt/apt.conf.d/99-no-pipelining
/etc/aws/
/etc/init.d/opsworks-agent
/etc/motd
/etc/motd.opsworks-static
/etc/sudoers.d/opsworks
/etc/sudoers.d/opsworks-agent
/etc/sysctl.d/70-opsworks-defaults.conf
/opt/aws/opsworks/
/usr/sbin/opsworks-agent-cli
/var/lib/aws/
/var/log/aws/
/vol/
```

File modificati

```
/etc/apt/apt.conf.d/99-no-pipelining
/etc/crontab
/etc/default/monit
/etc/group
/etc/gshadow
/etc/monit/monitrc
/etc/passwd
/etc/security/limits.conf (removing limits only for EC2 micro instances)
/etc/shadow
/etc/sudoers
```

La configurazione iniziale crea anche un file di scambio sulle EC2 micro istanze Amazon.

La configurazione iniziale apporta le seguenti modifiche ai sistemi Ubuntu.

Origini pacchetto

La configurazione iniziale cambia le origini del pacchetto come indicato di seguito.

- `deb http://archive.ubuntu.com/ubuntu/ ${code_name} main universe`

  Per: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name} main universe`
- `deb http://archive.ubuntu.com/ubuntu/ ${code_name}-updates main universe`

  Per: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name}-updates main universe`
- `deb http://archive.ubuntu.com/ubuntu ${code_name}-security main universe`

  Per: `deb-src http://archive.ubuntu.com/ubuntu ${code_name}-security main universe`
- `deb http://archive.ubuntu.com/ubuntu/ ${code_name}-updates multiverse`

  Per: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name}-updates multiverse`
- `deb http://archive.ubuntu.com/ubuntu ${code_name}-security multiverse`

  Per: `deb-src http://archive.ubuntu.com/ubuntu ${code_name}-security multiverse`
- `deb http://archive.ubuntu.com/ubuntu/ ${code_name} multiverse`

  Per: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name} multiverse`
- `deb http://security.ubuntu.com/ubuntu ${code_name}-security multiverse`

  Per: `deb-src http://security.ubuntu.com/ubuntu ${code_name}-security multiverse`

Pacchetti

La configurazione iniziale disinstalla `landscape` e installa i seguenti pacchetti.

`autofs`	`libicu-dev`	`libopenssl-ruby`

libssl-dev	libxml2-dev	libxslt-dev
libyaml-dev	monit	ntpd
procps	ruby	ruby-dev
rubygems	screen	sqlite
vim	xfs	

Modifica della configurazione dell'istanza

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi modificare le configurazioni delle istanze, incluse le istanze Amazon Elastic Compute Cloud EC2 (Amazon) registrate, con le seguenti limitazioni:

- L'istanza deve essere nello stato stopped (arrestato).

  Anche se non puoi modificare le proprietà di un'istanza online, puoi modificare alcuni aspetti della relativa configurazione modificando i livelli dell'istanza. Per ulteriori informazioni, consulta Modifica della configurazione di un livello OpsWorks .
- Alcune impostazioni, ad esempio Availability Zone (Zona di disponibilità) e Scaling Type (Tipo di dimensionamento), vengono definite quando crei l'istanza e non possono essere modificate in un secondo momento.
- Alcune impostazioni possono essere modificate, ad esempio solo per le istanze supportate da storage, non per le istanze supportate da Amazon Elastic Block Store.

  Ad esempio, puoi modificare il sistema operativo di un'istanza archiviata nell'archivio di un'istanza. Le istanze supportate da Amazon EBS devono utilizzare il sistema operativo specificato al

momento della creazione dell'istanza. Per ulteriori informazioni sullo storage dell'istanza, consulta Storage.

- Per impostazione predefinita, le istanze ereditano l'impostazione relativa alla versione dell'agente dello stack.

  Puoi utilizzare OpsWorks Agent Version per sovrascrivere l'impostazione della versione dell'agente dello stack e specificare una particolare versione dell'agente per un'istanza. Se specificate la versione dell'agente di un'istanza, AWS OpsWorks Stacks non aggiorna automaticamente l'agente quando è disponibile una nuova versione, anche se l'impostazione della versione dell'agente dello stack è Aggiornamento automatico. È necessario aggiornare manualmente la versione dell'agente dell'istanza modificando la configurazione dell'istanza. AWS OpsWorks Stacks installa quindi la versione dell'agente specificata sull'istanza.

> **Note**
>
> Non puoi modificare la configurazione di istanze locali registrate.

Per modificare la configurazione dell'istanza

1. Arrestare l'istanza in errore, se non è già stata arrestata.
2. Nella pagina Instances (Istanze), fare clic un nome di istanza per visualizzare la pagina Details (Dettagli).
3. Fare clic su Edit (Modifica) per visualizzare la pagina di modifica.
4. Modificare la configurazione dell'istanza, a seconda dei casi.

Per una descrizione delle impostazioni Host name (Nome host), Size (Dimensione), SSH key (Chiave SSH) e Operating system (Sistema operativo), consulta Aggiunta di un'istanza a un livello. L'impostazione Layers (Livelli) consente di aggiungere o rimuovere i livelli. Il livello corrente dell'istanza viene visualizzato dopo l'elenco di livelli.

- Per aggiungere un altro livello, selezionarlo nell'elenco.
- Per rimuovere l'istanza da uno dei relativi livelli, fare clic sull'icona x accanto al livello appropriato.

  Un'istanza deve essere membro di almeno un livello. Pertanto non è possibile eliminare l'ultimo livello.

Quando riavvii l'istanza, AWS OpsWorks Stacks avvia una nuova EC2 istanza Amazon con la configurazione aggiornata.

Eliminazione delle AWS OpsWorks istanze di Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi utilizzare AWS OpsWorks Stacks per interrompere un'istanza, incluse le istanze Amazon EC2 registrate. In questo modo si arresta l' EC2 istanza, ma l'istanza rimane nello stack. Puoi riavviarla facendo clic su start (avvio) nella colonna Actions (Operazioni) dell'istanza. Se non hai più bisogno di un'istanza e desideri rimuoverla dallo stack, puoi eliminarla, rimuovendo l'istanza dallo stack e terminando l'istanza Amazon associata. EC2 L'eliminazione di un'istanza elimina anche tutti i log o i dati associati e tutti i volumi Amazon Elastic Block Store (EBS) sull'istanza.

> **Important**
>
> Questo argomento si applica solo alle EC2 istanze Amazon gestite da AWS OpsWorks Stacks. Per ulteriori informazioni su come eliminare le istanze gestite dalla EC2 console o dall'API Amazon, consulta Terminate Your Instance.

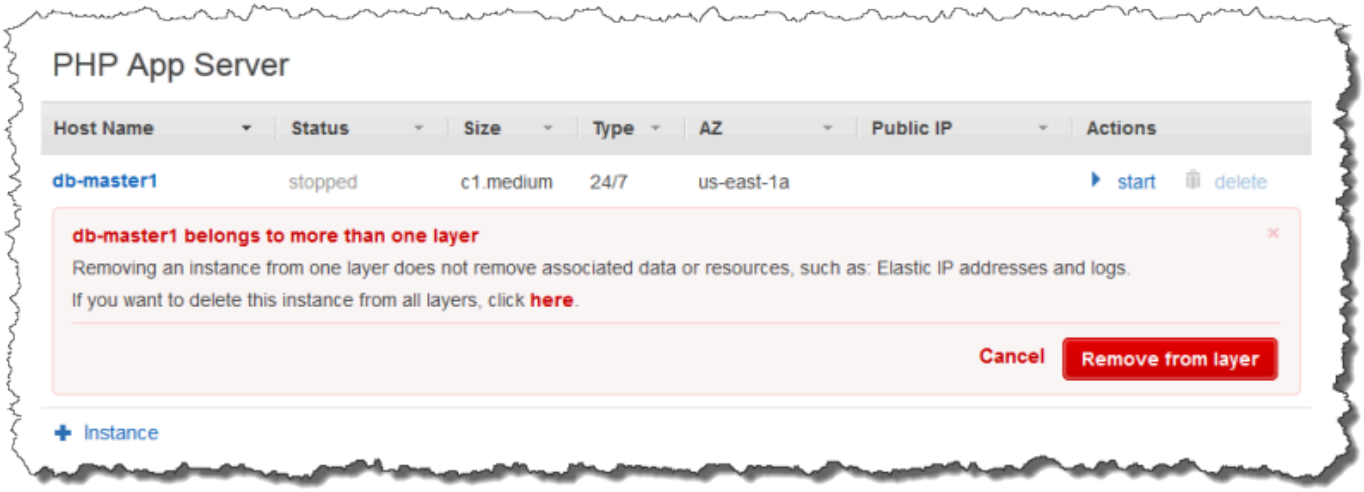
> **Note**
>
> Non puoi utilizzare AWS OpsWorks Stacks per eliminare un'istanza locale registrata.

Se un'istanza appartiene a più livelli, puoi eliminare l'istanza dallo stack oppure rimuovere solo un determinato livello. Puoi anche rimuovere i livelli dalle istanze modificando la configurazione dell'istanza, come descritto in Modifica della configurazione dell'istanza.

> **Important**
>
> È necessario eliminare le istanze AWS OpsWorks Stacks solo utilizzando la console o l'API AWS OpsWorks Stacks. In particolare, non dovresti eliminare le istanze AWS OpsWorks Stacks utilizzando la EC2 console o l'API Amazon perché EC2 le azioni Amazon non vengono sincronizzate automaticamente con Stacks. AWS OpsWorks Ad esempio, se la riparazione automatica è abilitata e si termina un'istanza utilizzando la EC2 console Amazon, AWS OpsWorks Stacks considera l'istanza terminata come un'istanza fallita e avvia un'altra istanza Amazon EC2 per sostituirla. Per ulteriori informazioni, consulta Utilizzo della diagnostica automatica.

Per eliminare un'istanza

1. Nella pagina Instances (Istanze), individuare l'istanza nel livello appropriato. Se l'istanza è in esecuzione, fare clic su stop (arresto) nella colonna Actions (Operazioni).
2. Dopo che lo stato diventa stopped (arrestato), fare clic su delete (eliminazione). Se l'istanza è membro di più di un livello, il layer AWS OpsWorks Stacks visualizza la sezione seguente.



- Per rimuovere l'istanza solo dal livello selezionato, fare clic su Remove from layer (Rimuovi dal livello).

  L'istanza rimane un membro dei relativi altri livelli e può essere riavviata.
- Per eliminare l'istanza da tutti i relativi livelli, rimuovendola pertanto dallo stack, fare clic su here (qui).

3. Se scegliete di rimuovere completamente un'istanza dallo stack o se l'istanza è membro di un solo livello, AWS OpsWorks Stacks vi chiede di confermare l'eliminazione.

   Seleziona Elimina per confermare. Oltre a eliminare l'istanza dallo stack, questa operazione elimina eventuali log o dati associati, nonché volume root collegati all'istanza. Per rimuovere tutti i volumi delle istanze, scegliere Elimina volumi EBS dell'istanza (gli snapshot non verranno eliminati) prima di scegliere Elimina.

Utilizzo di SSH per l'accesso a un'istanza Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi accedere alle tue istanze Linux online con SSH utilizzando il MindTerm client integrato o un client di terze parti, come PuTTY. SSH in genere dipende da una coppia di chiavi RSA per l'autenticazione. Si installa la chiave pubblica sull'istanza e si fornisce la chiave privata corrispondente al client SSH. AWS OpsWorks Stacks gestisce l'installazione delle chiavi pubbliche sulle istanze del tuo stack per te, come segue.

- Coppia di chiavi Amazon Elastic Compute Cloud (Amazon EC2): se la regione dello stack ha una o più coppie di EC2 chiavi Amazon, puoi specificare una coppia di chiavi SSH predefinita per lo stack.

  Puoi sostituire la coppia di chiavi predefinita e specificare una coppia di chiavi diversa al momento della creazione di un'istanza. In entrambi i casi, AWS OpsWorks Stacks installa la chiave pubblica della coppia di chiavi specificata sull'istanza. Per ulteriori informazioni su come creare coppie di EC2 chiavi Amazon, consulta Amazon EC2 Key Pairs.
- Coppia di chiavi personali: ogni utente può registrare una coppia di chiavi personale con AWS OpsWorks Stacks.

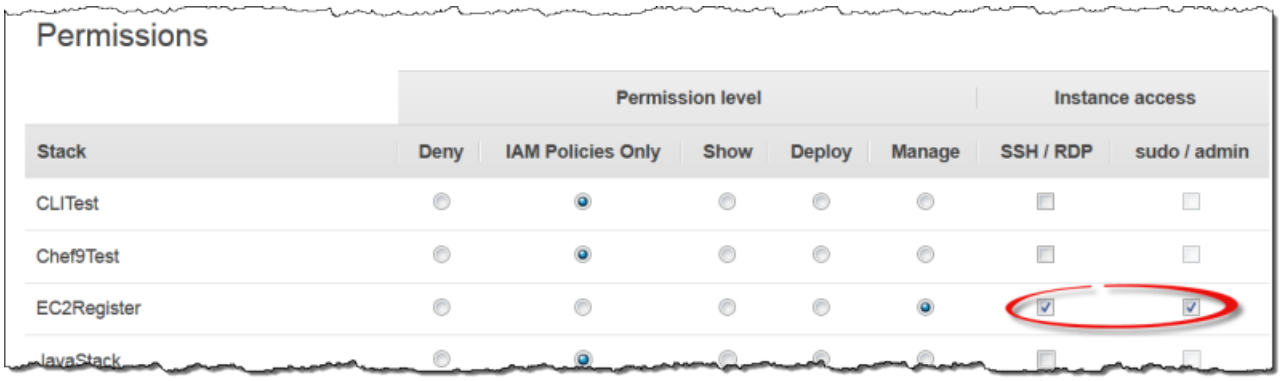
  L'utente o un amministratore registra la chiave pubblica con AWS OpsWorks Stacks e l'utente memorizza la chiave privata localmente. Quando si impostano le autorizzazioni per uno stack,

l'amministratore specifica gli utenti che devono avere un accesso SSH alle istanze dello stack. AWS OpsWorks Stacks crea automaticamente un utente di sistema sulle istanze dello stack per ogni utente autorizzato e installa la chiave pubblica personale degli utenti.

Un utente deve disporre dell'autorizzazione SSH per utilizzare il client MindTerm SSH o per utilizzare la propria key pair personale per accedere alle istanze di uno stack.

Per autorizzare SSH per un utente

1. Nel riquadro di navigazione AWS OpsWorks Stacks, fai clic su Autorizzazioni.
2. Seleziona SSH/RDP per l'utente IAM desiderato per concedere le autorizzazioni necessarie. Se desideri consentire all'utente di utilizzare **sudo** per elevare i privilegi, ad esempio per eseguire i comandi CLI dell'agente, seleziona anche sudo/admin.



Per ulteriori informazioni su come utilizzare Stacks per gestire l'accesso SSH, consulta. AWS OpsWorks Gestione dell'accesso SSH

Argomenti

- Utilizzo del client SSH integrato MindTerm
- Utilizzo di un client SSH di terze parti

Utilizzo del client SSH integrato MindTerm

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

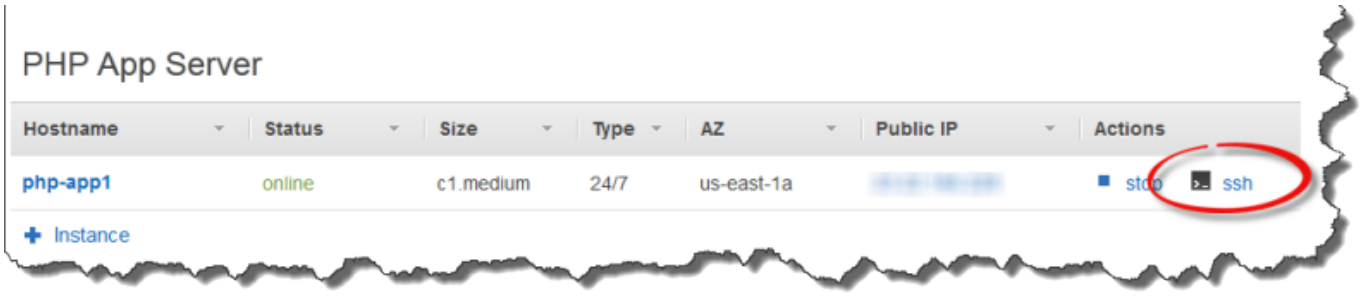
sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Il modo più semplice per accedere a un'istanza Linux è utilizzare il client MindTerm SSH integrato. Ogni istanza online include un'azione ssh che puoi usare per avviare il MindTerm client.

> **Note**
>
> È necessario che Java sia abilitato nel browser per utilizzare il MindTerm client.

Per accedere con il MindTerm client

1. Se necessario, autorizzare l'accesso SSH per l'utente IAM che deve connettersi all'istanza, come descritto nella sezione precedente.
2. Effettua il login come utente.
3. Nella pagina Instances (Istanze) scegliere ssh nella colonna Actions (Azioni) per l'istanza appropriata.



4. Per la chiave privata, fornisci un percorso alla chiave privata personale dell'utente o a una chiave EC2 privata Amazon, a seconda delle chiavi pubbliche installate sull'istanza.
5. Scegliere Launch Mindterm (Avvia Mindterm) e utilizzare la finestra del terminale per eseguire i comandi sull'istanza.

Utilizzo di un client SSH di terze parti

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi anche utilizzare un client SSH di terze parti, ad esempio PuTTY, per connetterti alle istanze Linux.

Per utilizzare un client SSH di terze parti

1. Assicurati che AWS OpsWorks Stacks abbia installato una chiave EC2 pubblica Amazon o una chiave pubblica personale di un utente IAM sull'istanza, come discusso in precedenza.
2. Ottenere il nome DNS pubblico dell'istanza o un indirizzo IP pubblico dalla relativa pagina dei dettagli.
3. Fornire al client il nome host dell'istanza, che dipende dal sistema operativo, come segue:

   - Amazon Linux e Red Hat Enterprise Linux (RHEL) — `ec2-user@`*`DNSName/Address`*`.`
   - Ubuntu —`ubuntu@`*`DNSName/Address`*.

   Sostituisci *`DNSName/Address`* con il nome DNS pubblico o l'indirizzo IP del passaggio precedente.
4. Fornire al client una chiave privata corrispondente a una chiave pubblica installata. Puoi utilizzare una chiave EC2 privata Amazon o la chiave privata personale di un utente IAM, a seconda delle chiavi pubbliche installate sull'istanza.

Utilizzo di RDP per l'accesso a un'istanza Windows

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi utilizzare Remote Desktop Protocol (RDP) di Windows per accedere a un'istanza Windows online nel modo descritto di seguito:

- L'istanza deve includere un gruppo di sicurezza con una regola per il traffico in entrata che permetta l'accesso RDP.

  Per ulteriori informazioni sull'utilizzo di gruppi di sicurezza, consulta Utilizzo dei gruppi di sicurezza.
- Utenti ordinari: AWS OpsWorks Stacks fornisce agli utenti ordinari autorizzati una password RDP valida per un periodo di tempo limitato, che può variare da 30 minuti a 12 ore.

  Oltre ad essere autorizzati, gli utenti devono disporre almeno di un livello di autorizzazione Show o le politiche associate AWS Identity and Access Management (IAM) devono consentire l'azione`opsworks:GrantAccess`.
- Amministratori: è possibile utilizzare la password dell'amministratore per accedere per un periodo di tempo illimitato.

  Come descritto più avanti, se hai specificato una coppia di chiavi Amazon Elastic Compute Cloud (Amazon EC2) per l'istanza, puoi usarla per recuperare la password dell'amministratore.

> **Note**
>
> Questo argomento descrive come utilizzare il client Connessione desktop remoto di Windows per accedere da una workstation Windows. Puoi anche utilizzare uno dei client RDP disponibili per Linux o OS X, ma la procedura potrebbe essere leggermente diversa. Per ulteriori informazioni sui client RDP compatibili con Microsoft Windows Server 2012 R2, consulta la pagina sui client Desktop remoto Microsoft.

Argomenti

- Assegnazione di un gruppo di sicurezza per permettere l'accesso RDP
- Accesso come utente normale
- Accesso come amministratore

Assegnazione di un gruppo di sicurezza per permettere l'accesso RDP

Prima di utilizzare RDP per accedere a un'istanza Windows, le regole del traffico in ingresso del gruppo di sicurezza dell'istanza devono permettere le connessioni RDP. Quando crei il primo stack

in una regione, AWS OpsWorks Stacks crea una serie di gruppi di sicurezza, Ne includono una denominata simile`AWS-OpsWorks-RDP-Server`, che AWS OpsWorks Stacks collega a tutte le istanze Windows per consentire l'accesso RDP. Tuttavia, per impostazione predefinita, questo gruppo di sicurezza non presenta regole, perciò devi aggiungere una regola in entrata per consentire l'accesso RDP alle istanze.

Per consentire l'accesso RDP

1. Apri la EC2console Amazon, impostala sulla regione dello stack e scegli Security Groups dal pannello di navigazione.
2. Seleziona AWS- OpsWorks -RDP-Server, scegli la scheda Inbound e scegli Modifica.
3. Scegliere Add Rule (Aggiungi regola) e specificare le seguenti impostazioni:
   - Tipo: RDP
   - Fonte: gli indirizzi IP di origine consentiti.

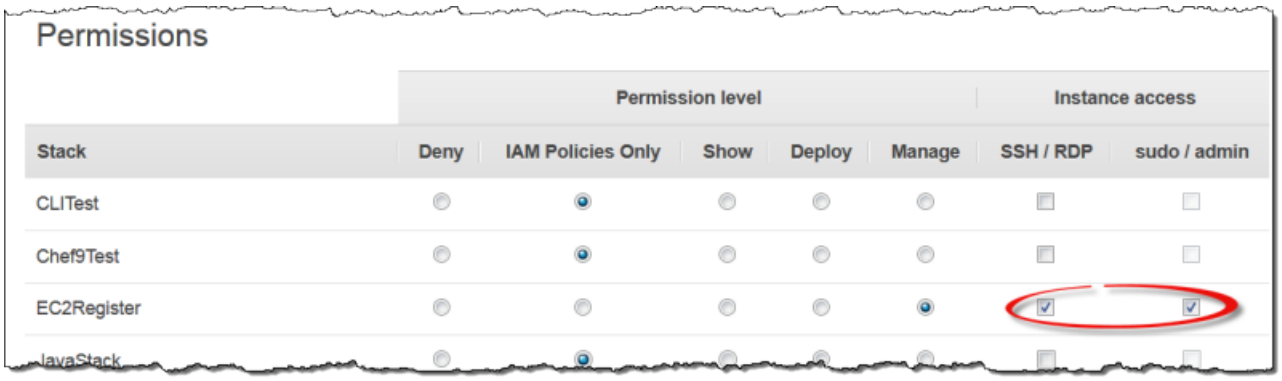
     In genere vengono permesse richieste RDP in entrata dall'indirizzo IP o da un intervallo di indirizzi IP specificato (di solito l'intervallo di indirizzi IP dell'azienda).

Accesso come utente normale

Un utente autorizzato può accedere alle istanze utilizzando una password temporanea, fornita da AWS OpsWorks Stacks.

Autorizzare RDP per un utente;

1. Nel riquadro di navigazione AWS OpsWorks Stacks, fai clic su Autorizzazioni.
2. Seleziona la casella di controllo SSH/RDP per l'utente desiderato per concedere le autorizzazioni necessarie. Se si desidera che l'utente abbia autorizzazioni di amministratore, è necessario selezionare anche sudo/admin .



Gli utenti autorizzati possono accedere a una qualsiasi delle istanze online dello stack, nel modo descritto di seguito.

Per accedere come utente IAM ordinario

1. Accedere come utente IAM.
2. Nella pagina Instances (Istanze) scegliere rdp nella colonna Actions (Operazioni) per l'istanza appropriata.
3. Specificare la durata della sessione, che può variare da 30 minuti a 12 ore, e scegliere Generate Password (Genera password). La password sarà valida solo per la durata della sessione specificata.
4. Registrare i valori di public DNS name (nome DNS pubblico), username (nome utente) e password e quindi scegliere Acknowledge and close (Riconosci e chiudi) .
5. Aprire il client Connessione desktop remoto di Windows, scegliere Mostra opzioni e immettere i valori seguenti dalle informazioni registrate nella fase 4:

   - Computer: il nome DNS pubblico dell'istanza.

     Se si preferisce, è anche possibile utilizzare l'indirizzo IP pubblico. Scegliere Istanze e copiare l'indirizzo indicato dalla colonna Public IP (IP pubblico) dell'istanza.
   - Nome utente: il nome utente.
6. Quando il client richiede le credenziali, immettere la password salvata nella fase 4.

> **Note**
>
> AWS OpsWorks Stacks genera una password utente solo per le istanze online. Se avvii un'istanza e, ad esempio, una delle ricette di impostazione personalizzate non riesce,

l'istanza otterrà lo stato `setup_failed`. Anche se l'istanza non è online per quanto riguarda AWS OpsWorks Stacks, l' EC2 istanza è in esecuzione ed è spesso utile accedere per risolvere il problema. AWS OpsWorks In questo caso Stacks non genererà una password per te, ma se hai assegnato una coppia di chiavi SSH all'istanza, puoi utilizzare la EC2 console o la CLI per recuperare la password dell'amministratore dell'istanza e accedere come amministratore. Per ulteriori informazioni, consulta la sezione seguente.

Accesso come amministratore

Puoi accedere come amministratore a un'istanza utilizzando la password appropriata. Se hai assegnato una EC2 key pair a un'istanza, Amazon la EC2 utilizza per creare e crittografare automaticamente una password di amministratore all'avvio dell'istanza. È quindi possibile utilizzare la chiave privata della coppia di chiavi con la EC2 console, l'API o la CLI per recuperare e decrittografare la password.

> **Note**
>
> Non è possibile utilizzare una coppia di chiavi SSH personale per recuperare una password di amministratore; è necessario utilizzare una coppia di EC2 chiavi.

Di seguito viene descritto come utilizzare la EC2 console per recuperare una password di amministratore e accedere a un'istanza. Se preferisci gli strumenti a riga di comando, puoi utilizzare anche il comando `get-password-data` dell'interfaccia a riga di comando di AWS per recuperare la password.

Per accedere come amministratore

1. Assicurati di aver specificato una EC2 key pair per l'istanza. È possibile specificare una coppia di chiavi predefinita per tutte le istanze dello stack al momento della creazione dello stack oppure specificare una coppia di chiavi per una determinata istanza al momento della creazione dell'istanza.
2. Apri la EC2 console, impostala sulla regione dello stack e scegli Istanze dal pannello di navigazione.
3. Selezionare l'istanza, scegliere Connect (Connetti)e quindi Get Password (Ottieni password).

4. Fornisci un percorso alla EC2 chiave privata della coppia di chiavi sulla tua workstation e scegli Decriptare la password. Copiare la password decrittografata per utilizzarla più avanti.
5. Aprire il client Connessione desktop remoto di Windows, scegliere Mostra opzioni e immettere le informazioni seguenti:
   - Computer: il nome DNS pubblico o l'indirizzo IP pubblico dell'istanza, che puoi ottenere dalla pagina dei dettagli dell'istanza.
   - Nome utente:`Administrator`.
6. Quando il client richiede le credenziali, immettere la password decrittografata nella fase 4.

App

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Un'app AWS OpsWorks Stacks rappresenta il codice che desideri eseguire su un server di applicazioni. Il codice stesso risiede in un repository come un archivio Amazon S3; l'app contiene le informazioni necessarie per distribuire il codice nelle istanze dell'application server appropriate.

Quando distribuisci un'applicazione, AWS OpsWorks Stacks attiva un evento Deploy, che esegue le ricette Deploy di ogni livello. AWS OpsWorks Stacks installa anche gli attributi di configurazione e distribuzione dello stack che contengono tutte le informazioni necessarie per distribuire l'app, come l'archivio dell'app e i dati di connessione al database.

Devi implementare ricette personalizzate che recuperano i dati di distribuzione dell'app dagli attributi di configurazione e distribuzione dello stack e gestiscono le attività di distribuzione.

Argomenti

- Aggiunta di app
- Distribuzione di app

- Modifica delle app
- Connessione di un'applicazione a un server di database
- Utilizzo delle variabili di ambiente
- Passaggio di dati alle applicazioni
- Utilizzo di chiavi SSH di repository Git
- Utilizzo di domini personalizzati
- Uso di SSL

Aggiunta di app

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

La prima fase del processo di distribuzione di un'applicazione nei server di applicazioni prevede l'aggiunta di un'app allo stack. L'app rappresenta l'applicazione e contiene una serie di metadati, ad esempio il nome e il tipo dell'applicazione, nonché le informazioni necessarie per distribuire l'applicazione sulle istanze del server, ad esempio l'URL dell'archivio. Devi disporre delle autorizzazioni Manage (Gestione) per aggiungere un'app a uno stack. Per ulteriori informazioni, consulta Gestione delle autorizzazioni utente.

> **Note**
>
> La procedura in questa sezione è valida per stack Chef 12 e stack di versioni più recenti. Per ulteriori informazioni su come aggiungere app ai livelli negli stack Chef 11, consulta Fase 2.4: Creazione e distribuzione di un'app - Chef 11.

Per aggiungere un'app a uno stack.

1. Inserisci il codice nel tuo repository preferito: un archivio Amazon S3, un repository Git, un repository Subversion o un archivio HTTP. Per ulteriori informazioni, consulta Origine dell'applicazione.
2. Fare clic su Apps (App) nel riquadro di navigazione. Nella pagina Apps (App) fare clic su Add an app (Aggiungi un'app) per la prima app. Per le app successive, fare clic su +App.
3. Utilizzare la pagina App New (Nuova app) per configurare l'app, come descritto nella sezione seguente.

Configurazione di un'app

La pagina Add App (Aggiungi app) è composta dalle seguenti sezioni: Settings (Impostazioni), Application source (Origine applicazione), Data Sources (Origini dati), Environment Variables (Variabili di ambiente), Add Domains (Aggiungi domini) e SSL Settings (Impostazioni SSL).

Argomenti

- Impostazioni
- Origine dell'applicazione
- Origini dati
- Variabili di ambiente
- Impostazioni relative a SSL e dominio

Impostazioni

Nome

Il nome dell'app, utilizzato per rappresentare l'app nell'interfaccia utente. AWS OpsWorks Stacks utilizza questo nome anche per generare un nome breve per l'app che viene utilizzato internamente e per identificare l'app negli attributi di configurazione e distribuzione dello stack. Dopo aver aggiunto l'app allo stack, per visualizzare il nome breve fare clic su Apps (App) nel riquadro di navigazione e quindi scegliere il nome dell'app per aprire la pagina dei dettagli.

Document root (Radice documento)

AWS OpsWorks Stacks assegna l'impostazione root del documento all'`[:document_root]`attributo negli attributi dell'app. `deploy` Il valore predefinito è `null`. Le

ricette di distribuzione possono recuperare tale valore dagli attributi `deploy` tramite la sintassi di nodo Chef standard e quindi distribuire il codice specificato nella posizione appropriata sul server. Per ulteriori informazioni su come distribuire le app, consulta Ricette di ditribuzione.

Origine dell'applicazione

Puoi distribuire app dai seguenti tipi di repository: Git, pacchetto Amazon S3, pacchetto HTTP e Altro. Tutti i tipi di archivio richiedono di specificare il tipo e l'URL dell'archivio. I singoli tipi di archivio hanno requisiti propri, come descritto di seguito.

> **Note**
>
> AWS OpsWorks Stacks distribuisce automaticamente le applicazioni dai repository standard ai livelli server integrati. Se utilizzi il tipo di repository Altro, che è l'unica opzione per gli stack di Windows, AWS OpsWorks Stacks inserisce le informazioni del repository negli `deploy`attributi dell'app, ma devi implementare ricette personalizzate per gestire le attività di distribuzione.

Argomenti

- Archivio HTTP
- Archivio Amazon S3
- Archivio Git
- Altri archivi

Archivio HTTP

Per usare come archivio un server HTTP accessibile pubblicamente:

1. Crea un archivio compresso (zip, gzip, bzip2, Java WAR o tarball) della cartella che contiene il codice dell'app e gli eventuali file associati.

   > **Note**
   >
   > AWS OpsWorks Stacks non supporta archivi tar non compressi.

2. Caricare il file di archivio sul server.

3. Per specificare l'archivio nella console, selezionare HTTP Archive (Archivio HTTP) come tipo di repository e immettere l'URL.

   Se l'archivio è protetto da password, in Application Source, specifica le credenziali di accesso.

Archivio Amazon S3

Per utilizzare un bucket Amazon Simple Storage Service come repository:

1. Crea un bucket Amazon S3 pubblico o privato. Per ulteriori informazioni, consulta la documentazione di Amazon S3.
2. AWS OpsWorks Affinché Stacks possa accedere ai bucket privati, devi essere un utente con almeno diritti di sola lettura sul bucket Amazon S3 e avrai bisogno dell'ID della chiave di accesso e della chiave di accesso segreta. Per ulteriori informazioni, consulta la documentazione di AWS Identity and Access Management.
3. Inserire il codice e i file associati in una cartella e archiviare la cartella in un archivio compresso (zip, gzip, bzip2, Java WAR o tarball).

   > **Note**
   >
   > AWS OpsWorks Stacks non supporta archivi tar non compressi.

4. Carica il file di archivio nel bucket Amazon S3 e registra l'URL.
5. Per specificare il repository nella console AWS OpsWorks Stacks, imposta il tipo di repository su S3 Archive e inserisci l'URL dell'archivio. Per un archivio privato, è necessario fornire anche l'ID di una chiave di accesso AWS e la chiave di accesso segreta la cui policy concede le autorizzazioni per accedere al bucket. Lasciare vuote queste impostazioni per gli archivi pubblici.

Archivio Git

Un repository Git fornisce il controllo del codice sorgente e il controllo delle versioni. AWS OpsWorks Stacks supporta siti di repository ospitati pubblicamente come GitHubBitbucket e server Git ospitati privatamente. Sia per le app che per i moduli Git secondari, il formato utilizzato per specificare l'URL dell'archivio in Application Source (Origine applicazione) dipende se l'archivio è pubblico o privato:

Archivio pubblico: utilizza i protocolli di sola lettura HTTPS o Git. Ad esempio, Nozioni di base sugli stack Linux Chef 11 utilizza un GitHub archivio pubblico a cui è possibile accedere tramite uno dei seguenti formati di URL:

- Protocollo Git di sola lettura: **`git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git`**
- HTTPS: **`https://github.com/amazonwebservices/opsworks-demo-php-simple-app.git`**

Archivio privato: utilizza il formato di lettura/scrittura SSH mostrato in questi esempi:

- Archivi Github: **`git@github.com:`*`project`*`/`*`repository`***.
- Archivi in un server Git: ***`user`*`@`*`server`*`:`*`project`*`/`*`repository`***

Se selezioni Git in Source Control (Controllo origine) vengono visualizzate altre due impostazioni opzionali:

Repository SSH key (Chiave SSH archivio)

Devi specificare una chiave SSH di distribuzione per accedere agli archivi Git privati. Questo campo richiede la chiave privata; la chiave pubblica è assegnata al repository Git. Per i moduli secondari Git, la chiave specificata deve avere accesso a tali moduli secondari. Per ulteriori informazioni, consulta Utilizzo di chiavi SSH di repository Git.

> **Important**
>
> La chiave SSH deploy non può richiedere una password; AWS OpsWorks Stacks non ha modo di passarla.

Branch/Revision (Ramo/Versione)

Se il repository ha più rami, AWS OpsWorks Stacks scarica il ramo principale per impostazione predefinita. Per specificare un ramo particolare, inserisci il nome del ramo, l' SHA1 hash o il nome del tag. Per specificare un determinato commit, immetti l'identificatore di commit completo a 40 cifre esadecimali.

Altri archivi

Se gli archivi standard non soddisfano i requisiti, puoi usare altri archivi, ad esempio Bazaar. Tuttavia, AWS OpsWorks Stacks non distribuisce automaticamente le app da tali repository. Devi

implementare le ricette personalizzati per gestire il processo di distribuzione e assegnarle agli eventi Deploy (Distribuzione) dei livelli appropriati. Per un esempio di come implementare le ricette di ditribuzione (Distribuzione), consulta Ricette di ditribuzione.

Origini dati

Questa sezione collega un database all'app. Sono disponibili le seguenti opzioni:

- RDS: collega uno dei livelli di servizio Amazon RDS dello stack.
- Nessuno: non collegare un server di database.

Se selezioni RDS, devi specificare le seguenti informazioni.

Istanza di database

L'elenco include tutti i livelli di servizio Amazon RDS. Puoi anche selezionare una delle seguenti opzioni:

(Obbligatorio) Specificare il server di database da collegare all'app. Il contenuto dell'elenco dipende dall'origine dati.

- RDS: un elenco dei livelli di servizio Amazon RDS dello stack.

Nome del database

(Opzionale) Specificare un nome di database.

- Livello Amazon RDS: inserisci il nome del database che hai specificato per l'istanza Amazon RDS.

  Puoi ottenere il nome del database dalla console Amazon RDS.

Quando distribuisci un'app con un database collegato, AWS OpsWorks Stacks aggiunge la connessione dell'istanza di database agli attributi dell'app. `deploy`

Puoi scrivere una ricetta personalizzata per recuperare le informazioni dagli attributi `deploy` e inserirle in un file a cui l'applicazione può accedere. Questo è l'unico modo per fornire informazioni sulla connessione al database all'applicazione di tipo Other (Altro).

Per ulteriori informazioni su come gestire le connessioni al database, consulta Connessione a un database.

Per scollegare un server di database da un'app, modifica la configurazione dell'app per specificare un server di database diverso oppure nessun server.

Variabili di ambiente

Per ogni app puoi specificare un set di variabili di ambiente specifiche per l'app. Ad esempio, se disponi di due applicazioni, le variabili di ambiente definite per la prima app non sono disponibili nella seconda app e viceversa. Puoi inoltre definire la stessa variabile di ambiente per più app e assegnarle un valore diverso per ogni applicazione.

> **Note**
>
> Non c'è un limite specifico al numero di variabili di ambiente. Tuttavia, la dimensione della struttura dati associata, che include i nomi, i valori e i valori dei flag protetti delle variabili, non può superare i 20 KB. Questo limite deve essere tale da poter far fronte la maggior parte, se non tutti i casi d'uso. Il superamento di tale limite potrebbe generare un errore del servizio (console) o un'eccezione (API) accompagnata dal messaggio "Environment: is too large (maximum is 20KB)" ("Variabile di ambiente troppo grande (massimo: 10 KB)").

AWS OpsWorks Stacks memorizza le variabili come attributi negli attributi dell'app. `deploy` Puoi impostare le ricette personalizzati in modo che recuperino tali valori utilizzando la sintassi di nodo standard di Chef. Per esempi di come accedere alle variabili di ambiente di un'app, consulta Utilizzo delle variabili di ambiente .

Chiave

Nome della variabile. Può contenere fino a 64 lettere maiuscole e minuscole, numeri e caratteri di sottolineatura (_), ma deve iniziare con una lettera o un carattere di sottolineatura.

Valore

Valore della variabile. Può contenere fino a 256 caratteri, che devono essere tutti stampabili.

Protected value (Valore protetto)

Indica se il valore è protetto. Questa impostazione consente di nascondere le informazioni sensibili, ad esempio le password. Se imposti Protected value (Valore protetto) per una variabile, dopo aver creato l'app:

- Nella pagina dei dettagli dell'app viene visualizzato solo il nome della variabile e non il valore.

- Se disponi dell'autorizzazione per modificare l'app, puoi fare clic su Update value (Aggiorna valore) per specificare un nuovo valore, ma non puoi visualizzare o modificare il valore precedente.

> **Note**
>
> I log di distribuzione Chef talvolta possono includere variabili di ambiente. Questo significa che nella console possono essere visualizzate variabili protette. Per evitare che le variabili protette vengano visualizzate nella console, ti consigliamo di utilizzare i bucket Amazon S3 come storage per le variabili protette che non desideri vengano visualizzate nella console. Un esempio di come utilizzare un bucket S3 a questo scopo è disponibile in Utilizzo di un bucket Amazon S3 in questa guida.

Impostazioni relative a SSL e dominio

Per il tipo di app Altro, AWS OpsWorks Stacks aggiunge le impostazioni agli attributi dell'app. `deploy` La ricette possono recuperare i dati da questi attributi e configurare il server in base alle esigenze.

Impostazioni relative al dominio

Questa sezione include un campo opzionale Add Domains (Aggiungi domini) per specificare i domini. Per ulteriori informazioni, consulta Utilizzo di domini personalizzati.

Impostazioni SSL

Questa sezione include un pulsante di alternanza SSL Support (Supporto SSL) che puoi utilizzare per abilitare o disabilitare il protocollo SSL. Se fai clic su Yes (Sì), dovrai fornire le informazioni sul certificato SSL. Per ulteriori informazioni, consulta Uso di SSL.

Distribuzione di app

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

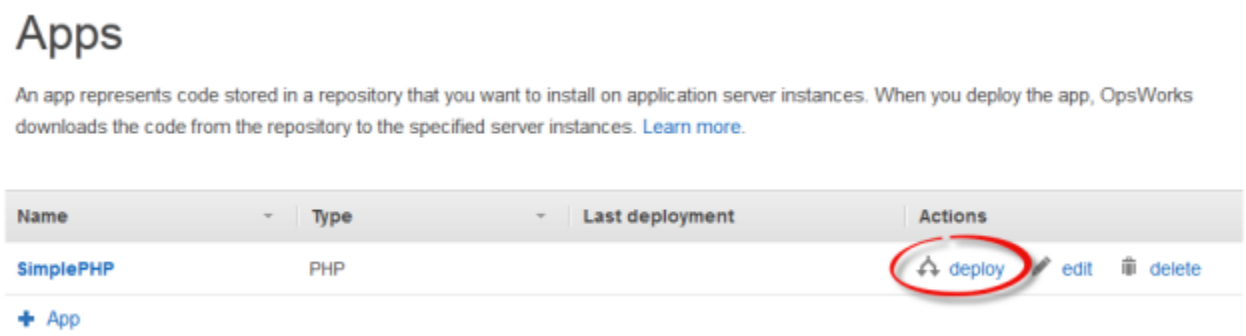
sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Lo scopo principale della distribuzione è distribuire il codice dell'applicazione e i file correlati in istanze del server applicazioni. L'operazione di distribuzione viene gestita dalle ricette di distribuzione di ogni istanza, determinate dal livello dell'istanza.

Quando avvii un'istanza, una volta completate le ricette di installazione, AWS OpsWorks Stacks esegue automaticamente le ricette Deploy dell'istanza. Tuttavia, quando aggiungi o modifichi un'app, devi distribuirla manualmente in qualsiasi istanza online. Per distribuire un'app, devi avere autorizzazioni di gestione o distribuzione. Per ulteriori informazioni, consulta Gestione delle autorizzazioni utente.

Per distribuire un'app

1. Nella pagina Apps (App) fare clic sull'operazione deploy (distribuzione) dell'app.



> **Note**
>
> È possibile distribuire un'app anche facendo clic su Deployments (Distribuzioni) nel riquadro di navigazione. Nella pagina Deployments & Commands (Distribuzioni e comandi) fare clic su Deploy an app (Distribuisci un'app). A questo punto, è possibile scegliere anche l'app da distribuire.

2. Specificare le impostazioni seguenti:

   - Obbligatorio: impostare Command (Comando) su deploy (distribuzione), se non è già selezionato.
   - Facoltativo: includere un commento.

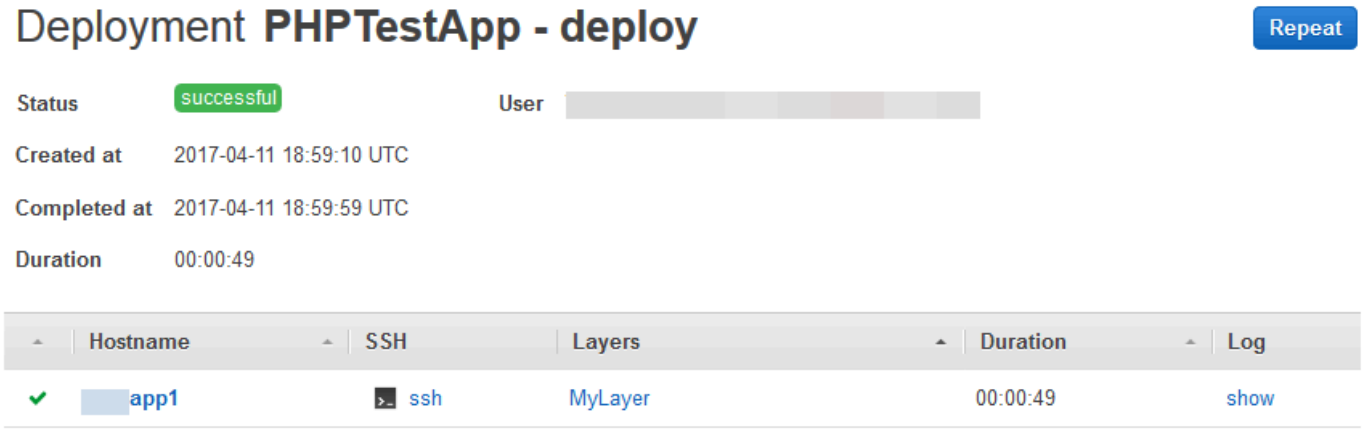
3. Fate clic su Avanzato >> per specificare un codice JSON personalizzato. AWS OpsWorks Stacks aggiunge un set di attributi di configurazione e distribuzione dello stack all'oggetto nodo. Gli attributi `deploy` contengono i dettagli della distribuzione e possono essere utilizzati da ricette di distribuzione per gestire l'installazione e la configurazione. Negli stack Linux, puoi utilizzare il campo JSON personalizzato per sovrascrivere le impostazioni AWS OpsWorks Stacks predefinite o passare impostazioni personalizzate alle tue ricette personalizzate. Per ulteriori informazioni su come utilizzare codice JSON personalizzato, consulta Utilizzo di un JSON personalizzato.

   > **Note**
   >
   > Se specifichi qui codice JSON personalizzato, questo viene aggiunto agli attributi di configurazione e distribuzione dello stack solo per questa distribuzione. Se vuoi aggiungere codice JSON personalizzato in modo permanente, devi aggiungerlo allo stack. Il codice JSON personalizzato è limitato a 120 KB. Se hai bisogno di maggiore capacità, ti consigliamo di archiviare alcuni dati su Amazon S3. Le ricette personalizzate possono quindi utilizzare l'interfaccia a riga di comando di AWS oppure l'SDK AWS per Ruby per scaricare i dati dal bucket all'istanza. Per vedere un esempio, consulta Utilizzo dell'SDK for Ruby.

4. In Instances (Istanze) fare clic su Advanced >> (Avanzate >>) e specificare le istanze in cui eseguire il comando di distribuzione.

   Il comando di distribuzione attiva un evento Deploy (Distribuzione), che esegue le ricette di distribuzione nelle istanze selezionate. Le ricette di distribuzione per il server applicazioni associato scaricano il codice e i file correlati dal repository e li installano nell'istanza e di conseguenza è normalmente possibile selezionare tutte le istanze del server applicazioni associate. Tuttavia, poiché altri tipi di istanza potrebbe richiedere alcune modifiche di configurazione per supportare la nuova app, è spesso utile eseguire le ricette di distribuzione anche in queste istanze. Le ricette aggiornano la configurazione nel modo appropriato, ma non installano i file dell'app. Per ulteriori informazioni sulle ricette, consulta Libri di ricette e ricette.

5. Fare clic su Deploy (Distribuisci) per eseguire le ricette di distribuzione nelle istanze specificate. Viene visualizzata la pagina Deployment (Distribuzione). Una volta completato il processo, AWS OpsWorks Stacks contrassegna l'app con un segno di spunta verde per indicare la corretta implementazione. Se la distribuzione non riesce, AWS OpsWorks Stacks contrassegna l'app con una X rossa. In tal caso, puoi andare alla pagina Distribuzioni ed esaminare il registro di distribuzione per ulteriori informazioni.



> **Note**
>
> Quando distribuisci un aggiornamento per un'app JSP, Tomcat potrebbe non riconoscere l'aggiornamento e continuare invece a eseguire la versione esistente dell'app. Questo può verificarsi, ad esempio, se distribuisci l'app come file ZIP contenente solo una pagina JSP. Per assicurarti che Tomcat esegua la versione distribuita più recente, la directory root del progetto deve includere una directory WEB-INF contenente un file `web.xml`. Un file `web.xml` può includere vari tipi di contenuto, ma quello riportato di seguito è sufficiente a garantire che Tomcat riconosca gli aggiornamenti ed esegua la versione dell'app attualmente distribuita. Non devi modificare la versione per ogni aggiornamento. Tomcat riconoscerà l'aggiornamento anche se la versione non è cambiata.
>
> ```
> <context-param>
>   <param-name>appVersion</param-name>
>   <param-value>0.1</param-value>
> </context-param>
> ```

Altri comandi di distribuzione

La pagina Deploy app (Distribuisci app) include diversi altri comandi per la gestione delle app e i server associati. Tra i comandi seguenti, solo Undeploy (Annulla distribuzione) è disponibile per app in stack Chef 12.

Undeploy (Annulla distribuzione)

Attiva un evento del ciclo di vita Undeploy (Annullamento distribuzione), che esegue ricette di annullamento della distribuzione per rimuovere tutte le versioni dell'app dalle istanze specificate.

Rollback

Ripristina la versione dell'app distribuita in precedenza. Ad esempio, se hai distribuito l'app tre volte e quindi esegui Rollback, il server renderà disponibile l'app della seconda distribuzione. Se esegui di nuovo Rollback, il server renderà disponibile l'app della prima distribuzione. Per impostazione predefinita, AWS OpsWorks Stacks archivia le cinque distribuzioni più recenti, il che consente di ripristinare fino a quattro versioni. Se superi il numero di versioni memorizzate, il comando non riesce e lascia la versione meno recente. Questo comando non è disponibile in stack Chef 12.

Start Web Server (Avvia server Web)

Esegue ricette che avviano il server applicazioni nelle istanze specificate. Questo comando non è disponibile in stack Chef 12.

Stop Web Server (Arresta server Web)

Esegue ricette che arrestano il server applicazioni nelle istanze specificate. Questo comando non è disponibile in stack Chef 12.

Restart Web Server (Riavvia server Web)

Esegue ricette che riavviano il server applicazioni nelle istanze specificate. Questo comando non è disponibile in stack Chef 12.

Important

Start Web Server (Avvia server Web), Stop Web Server (Arresta server Web), Restart Web Server (Riavvia server Web) e Rollback sono essenzialmente versioni personalizzate del comando dello stack Execute Recipes (Esegui ricette). Questi comandi eseguono un set di ricette che svolgono l'attività nelle istanze specificate.

- Poiché questi comandi non attivano un evento del ciclo di vita, non puoi impostarli per l'esecuzione di codice personalizzato.
- Questi comandi possono essere utilizzati solo per i livelli del server applicazioni predefiniti.

In particolare, non hanno effetto sui livelli personalizzati, anche se supportano un server applicazioni. Per avviare, arrestare o riavviare server in un livello personalizzato, devi implementare ricette personalizzate per l'esecuzione di queste attività e utilizzare il comando dello stack Execute Recipes (Esegui ricette) per eseguirle. Per ulteriori informazioni su come implementare e installare ricette personalizzate, consulta Libri di ricette e ricette.

Modifica delle app

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi modificare la configurazione di un'app mediante la modifica dell'app. Ad esempio, se sei pronto per distribuire una nuova versione, puoi modificare le impostazioni AWS OpsWorks Stacks dell'app per utilizzare il nuovo ramo del repository. Devi disporre delle autorizzazioni di gestione o distribuzione per poter modificare la configurazione di un'app. Per ulteriori informazioni, consulta Gestione delle autorizzazioni utente.

Per modificare un'app

1. Nella pagina Apps (App), fare clic sul nome dell'app per aprire la relativa pagina dei dettagli.
2. Fare clic su Edit (Modifica) per modificare la configurazione dell'app.

   - Se modifichi il nome dell'app, AWS OpsWorks Stacks utilizza il nuovo nome per identificare l'app nella console.

     La modifica del nome non comporta la modifica del nome breve associato. Il nome breve è impostato quando si aggiunge l'app allo stack e non può essere modificato in un secondo temo.

- Se si specifica una variabile di ambiente protetta, non è possibile visualizzare o modificare il valore. Tuttavia, è possibile specificare un nuovo valore facendo clic su Update value (Aggiorna valore).

3. Fare clic su Save (Salva) per salvare la nuova configurazione e quindi su Deploy App (Distribuisci app) per distribuire l'applicazione.

La modifica di un'app modifica le impostazioni con AWS OpsWorks Stacks, ma non influisce sulle istanze dello stack. Quando distribuisci un'app per la prima volta, le ricette di ditribuzione (Distribuzione) scaricano il codice e i file correlati nelle istanze del server di applicazioni, che quindi eseguiranno la copia locale. Se modifichi l'app nel repository o modifichi altre impostazioni, devi distribuire l'app per installare gli aggiornamenti sulle istanze del tuo app server, come segue. AWS OpsWorks Stacks distribuisce automaticamente la versione corrente dell'app su nuove istanze al momento dell'avvio. Per le istanze esistenti, tuttavia, la situazione è diversa:

- AWS OpsWorks Stacks distribuisce automaticamente la versione corrente dell'app su nuove istanze al momento dell'avvio.
- AWS OpsWorks Stacks distribuisce automaticamente la versione più recente dell'app sulle istanze offline, comprese le istanze basate sul caricamento e quelle basate sul tempo, al riavvio.
- Devi distribuire manualmente l'app aggiornata alle istanze online.

Per ulteriori informazioni su come distribuire le app, consulta Distribuzione di app.

Connessione di un'applicazione a un server di database

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi associare un server di database Amazon RDS a un'app al momento della creazione dell'app o successivamente modificandola. L'applicazione può quindi utilizzare le informazioni di connessione

al database: nome utente, password,... —per connettersi al server del database. Quando distribuisci un'app, AWS OpsWorks Stacks fornisce queste informazioni alle applicazioni in due modi:

- Per Linux stack, AWS OpsWorks Stacks crea un file su ciascuna delle istanze di server di applicazioni integrate che contengono dati di connessione che l'applicazione può utilizzare per connettersi al server di database.
- AWS OpsWorks Stacks include le informazioni di connessione negli attributi di configurazione e distribuzione dello stack installati su ciascuna istanza.

  È possibile implementare una ricetta personalizzata per estrarre le informazioni di connessione da questi attributi e metterla in un file nel formato desiderato. Per ulteriori informazioni, consulta Passaggio di dati alle applicazioni.

> **Important**
>
> Per gli stack Linux, se desideri associare un livello di servizio Amazon RDS alla tua app, devi aggiungere il pacchetto driver appropriato al livello di app server associato, come segue:
>
> 1. Fare clic su Layers (Livelli) nel riquadro di navigazione e aprire la scheda Recipes (Ricette) del server dell'applicazione.
> 2. Fare clic su Edit (Modifica) e aggiungere il pacchetto driver adeguato a OS Packages (Pacchetti OS). Ad esempio, è necessario specificare `mysql` se il livello contiene le istanze di Amazon Linux e `mysql-client` se il livello contiene istanze Ubuntu.
> 3. Salvare le modifiche e ridistribuire l'applicazione.

Utilizzo di una ricetta personalizzata

È possibile implementare una ricetta personalizzata che estrae i dati di connessione dagli `deploy` attributi dell'applicazione e li salva in un formato che l'applicazione è in grado di leggere, ad esempio come file YAML.

Si collega un server di database a un'applicazione quando si crea l'app o in seguito modifica l'app. Quando distribuisci l'app, AWS OpsWorks Stacks installa una configurazione dello stack e degli attributi di distribuzione su ogni istanza che includono le informazioni di connessione al database. L'applicazione può quindi recuperare gli attributi appropriati. I dettagli variano a seconda se si utilizza uno stack Linux o Windows.

Connessione a un server di database per un Linux Stack

Per gli stack Linux, lo spazio dei nomi di configurazione stack e attributi di distribuzione `deploy` includono un attributo per ogni applicazione distribuita, denominata con il nome breve dell'applicazione. Quando colleghi un server di database a un'app, AWS OpsWorks Stacks popola l'`[:database]`attributo dell'app con le informazioni di connessione e lo installa sulle istanze dello stack per ogni distribuzione successiva. I valori degli attributi vengono forniti dall'utente o generati da AWS OpsWorks Stacks.

> **Note**
>
> AWS OpsWorks Stacks consente di collegare un server di database a più app, ma ogni app può avere un solo server di database collegato. Se si desidera connettere un'applicazione a più di un server di database, collegare uno dei server all'applicazione e utilizzare le informazioni negli attributi `deploy` dell'app per connettersi al server. Utilizzare JSON personalizzato per inoltrare le informazioni di connessione per gli altri server di database all'applicazione. Per ulteriori informazioni, consulta Passaggio di dati alle applicazioni.

Un'applicazione può utilizzare le informazioni di connessione dagli attributi `deploy` dell'istanza per connettersi a un database. Tuttavia, le applicazioni non possono accedere direttamente a tali informazioni: solo le ricette possono accedere agli attributi. `deploy` È possibile affrontare questo problema implementando una ricetta personalizzata che estrae le informazioni di connessione dagli attributi `deploy` e le inserisce in un file che può essere letto dall'applicazione.

Utilizzo delle variabili di ambiente

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> I suggerimenti riportati in questo argomento si applicano a Chef 11.10 e versioni precedenti di Chef. Per ottenere le variabili di ambiente in Chef 12 e nelle versioni più recenti, devi utilizzare il contenitore di dati dell'app. Per ulteriori informazioni, consulta AWS OpsWorks Data Bag Reference e App Data Bag (aws_opsworks_app).

Quando specifichi le variabili di ambiente per un'app, AWS OpsWorks Stacks aggiunge le definizioni delle variabili agli attributi dell'app. `deploy`

I livelli personalizzati possono utilizzare una ricetta per recuperare il valore di una variabile utilizzando la sintassi standard di nodo e archiviarlo in un formato accessibile alle app del livello.

Devi implementare una ricetta personalizzata in grado di recuperare i valori delle variabili di ambiente dagli attributi `deploy` dell'istanza. La ricetta può quindi archiviare i dati nell'istanza in un formato accessibile dall'applicazione, ad esempio un file YAML. Le definizioni delle variabili di ambiente di un'app vengono archiviate negli attributi `deploy`, all'interno della variabile `environment_variables` dell'app. L'esempio seguente mostra il percorso di questi attributi per un'app denominata `simplephpapp`, utilizzando dati JSON per rappresentare la struttura degli attributi.

```
{
  ...
  "ssh_users": {
  },
  "deploy": {
    "simplephpapp": {
      "application": "simplephpapp",
      "application_type": "php",
      "environment_variables": {
        "USER_ID": "168424",
        "USER_KEY": "somepassword"
      },
    ...
  }
}
```

Una ricetta può recuperare i valori delle variabili utilizzando una sintassi standard di nodo. L'esempio seguente mostra come recuperare il valore `USER_ID` dal precedente dato JSON e inserirlo nel log di Chef.

```
Chef::Log.info("USER_ID: #{node[:deploy]['simplephpapp'][:environment_variables]
[:USER_ID]}")
```

Per una descrizione più dettagliata di come recuperare informazioni dai dati JSON di configurazione e distribuzione dello stack e archiviarle nell'istanza, consulta Passaggio di dati alle applicazioni.

Passaggio di dati alle applicazioni

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Spesso risulta utile passare dati quali, ad esempio, coppie chiave-valore a un'applicazione nel server. A tale scopo, devi utilizzare dati JSON personalizzati per aggiungere i dati allo stack. AWS OpsWorks Stacks aggiunge i dati all'oggetto nodo di ogni istanza per ogni evento del ciclo di vita.

Si noti, tuttavia, che anche se le ricette possono recuperare i dati JSON personalizzati dall'oggetto nodo tramite gli attributi di Chef, le applicazioni non sono in grado di eseguire questa operazione. Un approccio che consenta di passare i dati JSON personalizzati a una o più applicazioni è implementare una ricetta personalizzata in grado di estrarre i dati dall'oggetto `node` e quindi di scriverli in un file che l'applicazione è in grado di leggere. L'esempio utilizzato in questo argomento illustra come scrivere i dati a un file YAML. Puoi tuttavia utilizzare lo stesso approccio di base per altri formati, ad esempio JSON o XML.

Per passare i dati chiave-valore alle istanze dello stack, allo stack devi aggiungere dati JSON personalizzati, come quelli riportati di seguito. Per ulteriori informazioni su come aggiungere dati JSON personalizzati allo stack, consulta Utilizzo di un JSON personalizzato.

```
{
```

```
  "my_app_data": {
    "app1": {
      "key1": "value1",
      "key2": "value2",
      "key3": "value3"
    },
    "app2": {
      "key1": "value1",
      "key2": "value2",
      "key3": "value3"
    }
  }
}
```

L'esempio presuppone la presenza di due app i cui nomi brevi sono `app1` e `app2`, ciascuna delle quali include tre valori di dati. La ricetta corrispondente presuppone l'utilizzo dei nomi brevi delle app per identificare i dati associati; gli altri nomi sono arbitrari. Per ulteriori informazioni sui nomi brevi delle app, consulta Impostazioni.

La ricetta nell'esempio seguente mostra come estrarre i dati per ogni app dagli attributi `deploy` e come inserirli in un file `.yml`. La ricetta presuppone che i dati JSON personalizzati contengano dati per ogni applicazione.

```
node[:deploy].each do |app, deploy|
  file File.join(deploy[:deploy_to], 'shared', 'config', 'app_data.yml') do
    content YAML.dump(node[:my_app_data][app].to_hash)
  end
end
```

Gli attributi `deploy` includono un attributo per ogni app, denominato utilizzando il nome breve dell'app. Ogni attributo dell'app contiene un set di attributi che rappresentano una serie di informazioni relative all'app. In questo esempio viene utilizzata la directory di distribuzione dell'app, che è rappresentata dall'attributo `[:deploy][:app_short_name][:deploy_to]`. Per ulteriori informazioni su `[:deploy]`, consulta Attributi deploy.

Per ogni app in `deploy`, la ricetta esegue le seguenti operazioni:

1. Crea un file denominato `app_data.yml` nella sottodirectory `shared/config` della directory `[:deploy_to]`dell'applicazione.

Per ulteriori informazioni su come AWS OpsWorks Stacks installa le app, consulta. Ricette di ditribuzione

2. Converte i valori JSON personalizzati dell'app nel formato YAML e scrive i dati formattati in `app_data.yml`.

Per passare i dati a un'app

1. Aggiungere un'app allo stack e annotare il relativo nome breve. Per ulteriori informazioni, consulta Aggiunta di app.
2. Aggiungere dati JSON personalizzati contenenti i dati dell'app agli attributi `deploy`, come descritto in precedenza. Per ulteriori informazioni su come aggiungere dati JSON personalizzati allo stack, consulta Utilizzo di un JSON personalizzato.
3. Creare un libro di ricette e aggiungervi una ricetta contenente il codice riportato nell'esempio precedente, modificato in base alle esigenze relativamente ai nomi di attributo utilizzati nei dati JSON personalizzati. Per ulteriori informazioni su come creare libri di ricette e ricette, consulta Libri di ricette e ricette. Se per questo stack sono già disponibili libri di ricette personalizzati, è possibile anche aggiungere la ricetta a un libro di ricette esistente oppure aggiungere il codice a una ricetta Deploy (Distribuzione) esistente.
4. Installare il libro di ricette sullo stack. Per ulteriori informazioni, consulta Installazione di libri di ricette personalizzati.
5. Assegna la ricetta all'evento Deploy lifecycle del livello del server dell'app. AWS OpsWorks Stacks eseguirà quindi la ricetta su ogni nuova istanza, dopo l'avvio. Per ulteriori informazioni, consulta Esecuzione di ricette.
6. Distribuire l'app. Questa operazione installa anche gli attributi di configurazione e distribuzione dello stack, che ora contengono i dati.

> **Note**
>
> Se i file di dati devono essere disponibili prima della distribuzione dell'app, è anche possibile assegnare la ricetta all'evento Setup (Configurazione) del ciclo di vita del livello. Questo evento si verifica una volta, al completamento dell'avvio dell'istanza. Tuttavia, AWS OpsWorks Stacks non avrà ancora creato le directory di distribuzione, quindi la ricetta dovrebbe creare le directory richieste in modo esplicito prima di creare il file di dati. L'esempio

seguente crea in modo esplicito la directory `/shared/config` dell'app, quindi crea un file di dati in tale directory.

```
node[:deploy].each do |app, deploy|

 directory "#{deploy[:deploy_to]}/shared/config" do
      owner "deploy"
      group "www-data"
      mode 0774
      recursive true
      action :create
    end

  file File.join(deploy[:deploy_to], 'shared', 'config', 'app_data.yml') do
    content YAML.dump(node[:my_app_data][app].to_hash)
  end
end
```

Per caricare i dati, puoi utilizzare il seguente codice Sinatra:

```
#!/usr/bin/env ruby
# encoding: UTF-8
require 'sinatra'
require 'yaml'

get '/' do
  YAML.load(File.read(File.join('..', '..', 'shared', 'config', 'app_data.yml')))
End
```

Puoi aggiornare i valori dei dati dell'app in qualsiasi momento mediante l'aggiornamento dei dati JSON personalizzati, come descritto di seguito.

Per aggiornare i dati dell'applicazione

1. Modificare i dati JSON personalizzati per aggiornare i valori dei dati.
2. Distribuisci nuovamente l'app, che indica a AWS OpsWorks Stacks di eseguire le ricette Deploy sulle istanze dello stack. Le ricette utilizzerà gli attributi derivati dagli attributi di configurazione e

distribuzione aggiornati dello stack. Pertanto la ricetta personalizzata aggiornerà i file di dati con i valori correnti.

Utilizzo di chiavi SSH di repository Git

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

La chiave SSH di un repository Git, chiamata talvolta chiave SSH di distribuzione, è una chiave SSH senza password che consente di accedere a un repository Git privato. Idealmente, non appartiene ad alcuno sviluppatore specifico. Il suo scopo è consentire a AWS OpsWorks Stacks di distribuire in modo asincrono app o libri di cucina da un repository Git senza richiedere ulteriori input da parte dell'utente.

Di seguito viene descritta la procedura di base per la creazione della chiave SSH di un repository. Per ulteriori informazioni, consulta la documentazione relativa al repository. Ad esempio, Managing deploy keys descrive come creare una chiave SSH di repository per un repository, mentre Deployment Keys on Bitbucket descrive come creare una chiave SSH di GitHub repository per un repository Bitbucket. Un documento descrive la creazione di una chiave su un server. Per AWS OpsWorks Stacks, basta sostituire «server» con «workstation» nelle istruzioni.

Per creare una chiave SSH di repository

1. Creare una coppia di chiavi SSH di distribuzione per il repository Git sulla workstation utilizzando un programma, ad esempio `ssh-keygen`.

   > **Important**
   >
   > AWS OpsWorks Stacks non supporta le passphrase delle chiavi SSH.

2. Assegnare la chiave pubblica al repository e archiviare la chiave privata sulla workstation.

3. Immettere la chiave privata nella casella Repository SSH Key (Chiave SSH repository) quando si aggiunge un'app o si specifica il repository di un libro di ricette. Per ulteriori informazioni, consulta Aggiunta di app.

AWS OpsWorks Stacks passa la chiave SSH del repository a ciascuna istanza e le ricette integrate utilizzano quindi la chiave per connettersi al repository e scaricare il codice. La chiave viene memorizzata negli attributi `deploy` come `node[:deploy]['appshortname'][:scm][:ssh_key]` ed è accessibile solo all'utente root.

Utilizzo di domini personalizzati

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se ospiti il nome di un dominio di una parte terza, puoi mappare quel nome di dominio per un'app. La seguente è la procedura di base:

1. Crea un sottodominio con il registar DNS e mappalo sull'indirizzo IP elastico del tuo sistema di bilanciamento del carico, oppure sull'indirizzo IP pubblico del server dell'app.
2. Aggiorna la configurazione dell'app per scegliere il sottodominio e ridistribuire l'app.

> **Note**
>
> Assicurati di inoltrare il tuo nome di dominio non qualificato (ad esempio myapp1.example.com) a un nome di dominio qualificato (ad esempio www.myapp1.example.com), affinché entrambi mappino la tua app.

Quando configuri un dominio per un'app, viene elencato come un alias del server nel file di configurazione del server. Se stai utilizzando un sistema di bilanciamento del carico, questo controlla il nome di dominio nell'URL quando arrivano richieste e reindirizza il traffico in base al dominio.

Per mappare un sottodominio a un indirizzo IP

1. Se stai utilizzando un sistema di bilanciamento del carico, nella pagina Instances (Istanze), fai clic sull'istanza del sistema di bilanciamento del carico per aprire la pagina dei suoi dettagli e per ottenere l'indirizzo Elastic IP (IP elastico) dell'istanza. In caso contrario, ottieni l'indirizzo IP pubblico dalla pagina dei dettagli dell'istanza del server dell'applicazione.
2. Segui le istruzioni fornite dal registrar DNS per creare e mappare il tuo sottodominio per l'indirizzo IP dalla Fase 1.

> **Note**
>
> Se l'istanza del sistema di bilanciamento del carico a un certo punto viene terminata, ti verrà assegnato un nuovo indirizzo IP elastico. È necessario aggiornare le impostazioni del tuo registar DNS per mappare il nuovo indirizzo IP elastico.

AWS OpsWorks Stacks aggiunge semplicemente le impostazioni del dominio agli attributi dell'app. `deploy` È necessario implementare una ricetta personalizzata per recuperare le informazioni dall'oggetto nodo e configurare il server in modo appropriato. Per ulteriori informazioni, consulta Libri di ricette e ricette.

Esecuzione di più applicazioni sullo stesso server dell'applicazione

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Le informazioni incluse in questo argomento non sono valide per le app Node.js.

Se disponi di più applicazioni dello stesso tipo, talvolta è più conveniente eseguirle sulle stesse istanze del server dell'applicazione.

Per eseguire più applicazioni sullo stesso server

1. Aggiungi un'app allo stack di ogni applicazione.
2. Ottieni un sottodominio separato per ogni app e mappa i sottodomini per i server dell'applicazione o l'indirizzo IP di un sistema di bilanciamento del carico.
3. Modifica la configurazione di ogni app per specificare il sottodominio appropriato.

Per ulteriori informazioni su come eseguire queste attività, consulta Utilizzo di domini personalizzati.

> **Note**
>
> Se i server delle applicazioni eseguono più applicazioni HTTP, puoi utilizzare Elastic Load Balancing per il bilanciamento del carico. Per più applicazioni HTTPS, devi terminare la connessione SSL al sistema di bilanciamento del carico o creare uno stack separato per ogni applicazione. Le richieste HTTPS sono crittografate, il che significa che se interrompi la connessione SSL al server, il sistema di bilanciamento del carico non è in grado di verificare il nome di dominio per determinare quale applicazione deve gestire la richiesta.

Uso di SSL

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per utilizzare SSL con l'applicazione, è necessario prima ottenere un certificato server digitale da una Certificate Authority (CA). Per semplicità, questa procedura guidata crea un certificato poi lo autofirma. I certificati autofirmati sono utili per scopi di formazione ed esecuzione di test, ma è necessario utilizzare sempre un certificato firmato da una CA per gli stack di produzione.

In questa procedura guidata, si eseguiranno le operazioni seguenti:

1. Installare e configurare OpenSSL
2. Crea una chiave privata.
3. Creare una richiesta di firma del certificato.
4. Generare un certificato autofirmato.
5. Modificare l'applicazione con le informazioni del certificato.

Important

Se la tua applicazione utilizza SSL, ti consigliamo di disabilitarlo SSLv3, se possibile, nei livelli del server delle applicazioni per risolvere le vulnerabilità descritte in CVE-2014-3566. Se lo stack include un livello Ganglia, è necessario disabilitare SSL v3 anche per quel livello. I dettagli dipendono dal livello specifico; per ulteriori informazioni, consulta quanto segue.

- Java App Server AWS OpsWorks Stacks Layer
- Node.js App Server AWS OpsWorks Stacks Layer
- PHP App Server AWS OpsWorks Stacks Layer
- Rails AWS OpsWorks App Server Stacks Layer
- Static Web Server AWS OpsWorks Stacks Layer
- Ganglia Layer

Argomenti

- Passaggio 1: installare e configurare OpenSSL
- Passaggio 2: creare una chiave privata
- Passaggio 3: creare una richiesta di firma del certificato
- Passaggio 4: inviare il CSR alla Certificate Authority
- Passaggio 5: modificare la app

Passaggio 1: installare e configurare OpenSSL

La creazione e il caricamento dei certificati server richiede uno strumento che supporta i protocolli SSL e TLS. OpenSSL è uno strumento open-source che fornisce le funzioni di crittografia di base necessarie per creare un token RSA e firmarlo con la chiave privata.

La procedura seguente presuppone che il computer non disponga già di OpenSSL installato.

Per installare OpenSSL su Linux e Unix

1. Andare su OpenSSL: Source, Tarballs (OpenSSL: origine, tarball).
2. Scaricare l'origine più recente.
3. Creare il pacchetto.

Per installare OpenSSL in ambiente Windows

1. Se il Microsoft Visual C++ 2008 Redistributable Package non è già installato sul sistema, scarica il pacchetto.
2. Eseguire il programma di installazione e seguire le istruzioni fornite da Microsoft Visual C++ 2008 Redistributable Setup Wizard per installare il ridistribuibile.
3. Andare su OpenSSL: Binary Distributions (OpenSSL: distribuzione binaria), fare clic sulla versione appropriata dei file binari OpenSSL per l'ambiente e salvare il programma di installazione in locale.
4. Eseguire il programma di installazione e seguire le istruzioni in OpenSSL Setup Wizard (Procedura guidata di installazione OpenSSL) per installare il file binari.

Creare una variabile di ambiente che punti al punto di installazione OpenSSL attraverso l'apertura di un terminale o di una finestra di comando e l'utilizzo delle seguenti righe di comando.

- Su Linux e Unix

```
export OpenSSL_HOME=path_to_your_OpenSSL_installation
```

- Su Windows

```
set OpenSSL_HOME=path_to_your_OpenSSL_installation
```

Aggiungere la variabile di percorso dei file binari OpenSSL sul computer tramite l'apertura di un terminale o di una finestra di comando e l'utilizzo delle seguenti righe di comando.

- Su Linux e Unix

```
export PATH=$PATH:$OpenSSL_HOME/bin
```

- Su Windows

```
set Path=OpenSSL_HOME\bin;%Path%
```

> **Note**
>
> Eventuali modifiche apportate alle variabili di ambiente tramite l'utilizzo di queste righe di comando sono valide solo per la sessione della riga di comando attuale.

Passaggio 2: creare una chiave privata

È necessaria una chiave privata univoca per creare la richiesta di firma del certificato (CSR). Crea la chiave utilizzando la seguente riga di comando:

```
openssl genrsa 2048 > privatekey.pem
```

Passaggio 3: creare una richiesta di firma del certificato

Una richiesta di firma del certificato (CSR) è un file inviato a una Certificate Authority (CA) per richiedere un certificato server digitale. Creare la CSR utilizzando la seguente riga di comando.

```
openssl req -new -key privatekey.pem -out csr.pem
```

L'output del comando risulterà simile al seguente:

```
You are about to be asked to enter information that will be incorporated
 into your certificate request.
 What you are about to enter is what is called a Distinguished Name or a DN.
 There are quite a few fields but you can leave some blank
```

```
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

La tabella seguente aiuta a creare la richiesta di certificato.

Dati di richiesta di certificato

Nome	Descrizione	Esempio
Nome paese	L'abbreviazione ISO di due lettere per il tuo paese.	US = Stati Uniti
Lo stato o la provincia	Il nome dello stato o della provincia in cui si trova la tua organizzazione. Questo nome non può essere abbreviato.	Washington
Locality Name (Nome località)	Il nome della città in cui si trova la tua organizzazione.	Seattle
Nome organizzazione	La denominazione legale completa della tua organizza zione. Non abbreviare il nome dell'organizzazione.	CorporationX
Unità organizzativa	(Opzionale) Per informazioni aggiuntive sull'organizzazione.	Marketing
Common Name (Nome comune)	Il nome di dominio pienament e qualificato per il CNAME. Si riceverà un avviso di controllo del nome di certificato se non è una corrispondenza esatta.	www.example.com
Indirizzo e-mail	L'indirizzo e-mail dell'ammi nistratore del server	someone@example.com

> **Note**
>
> Il campo Nome comune è spesso frainteso e viene completato in modo non corretto. Il nome comune è in genere l'host più il nome di dominio. Comparirà come "www.example.com" o "example.com". È necessario creare una CSR utilizzando il proprio nome comune corretto.

Passaggio 4: inviare il CSR alla Certificate Authority

Per l'utilizzo di produzione, è possibile ottenere un certificato server inviando il CSR a una Certificate Authority (CA) che potrebbe richiedere altre credenziali o prove di identità. Se la richiesta va a buon fine, la CA restituisce il certificato di identità con la firma digitale e possibilmente un file della catena del certificato. AWS non consiglia una CA specifica. Per un elenco parziale dei prodotti disponibili CAs, consulta Certificate Authority - Providers su Wikipedia.

È inoltre possibile generare un certificato autofirmato che può essere utilizzato solo a scopo di test. Per questo esempio, utilizzare la seguente riga di comando per generare un certificato autofirmato.

```
openssl x509 -req -days 365 -in csr.pem -signkey privatekey.pem -out server.crt
```

L'output risulterà simile al seguente:

```
Loading 'screen' into random state - done
Signature ok
subject=/C=us/ST=washington/L=seattle/O=corporationx/OU=marketing/CN=example.com/
emailAddress=someone@example.com
Getting Private key
```

Passaggio 5: modificare la app

Dopo aver generato il certificato e averlo firmato, aggiornare la app per attivare SSL e fornire le informazioni di certificato. Nella pagina App scegliere un'app per aprire la pagina dei dettagli e fare clic su Edit App (Modifica app). Per attivare il supporto SSL, impostare Enable SSL (Abilita SSL) su Sì per visualizzare le seguenti opzioni di configurazione.

Certificato SSL

Copiare i contenuti del file di certificato della chiave pubblica (.crt) nella casella. Il certificato deve avere un aspetto simile al seguente:

```
-----BEGIN CERTIFICATE-----
MIICuTCCAiICCQCtqFKItVQJpzANBgkqhkiG9w0BAQUFADCBoDELMAkGA1UEBhMC
dXMxEzARBgNVBAgMCndhc2hpbmd0b24xEDAOBgNVBAcMB3NlYXR0bGUxDzANBgNV
BAoMBmFtYXpvbjEWMBQGA1UECwwNRGV2IGFuZCBUb29sczEdMBsGA1UEAwwUc3Rl
cGhhbmllYXBpZXJjZS5jb20xIjAgBgkqhkiG9w0BCQEWE3NhcGllcmNlQGFtYXpv
...
-----END CERTIFICATE-----
```

> **Note**
>
> Se si sta utilizzando Nginx e si dispone di un file della catena del certificato, è necessario aggiungere i contenuti al file di certificato della chiave pubblica.

Se si sta aggiornando un certificato esistente, eseguire le operazioni seguenti:

- Scegliere Update SSL certificate (Aggiorna certificato SSL) per aggiornare il certificato.
- Se il nuovo certificato non corrisponde alla chiave privata esistente, scegliere Update SSL certificate key (Aggiorna chiave certificato SSL).
- Se il nuovo certificato non corrisponde alla catena di certificati esistente, scegliere Update SSL certificates (Aggiorna certificato SSL).

SSL Certificate Key (Chiave del certificato SSL)

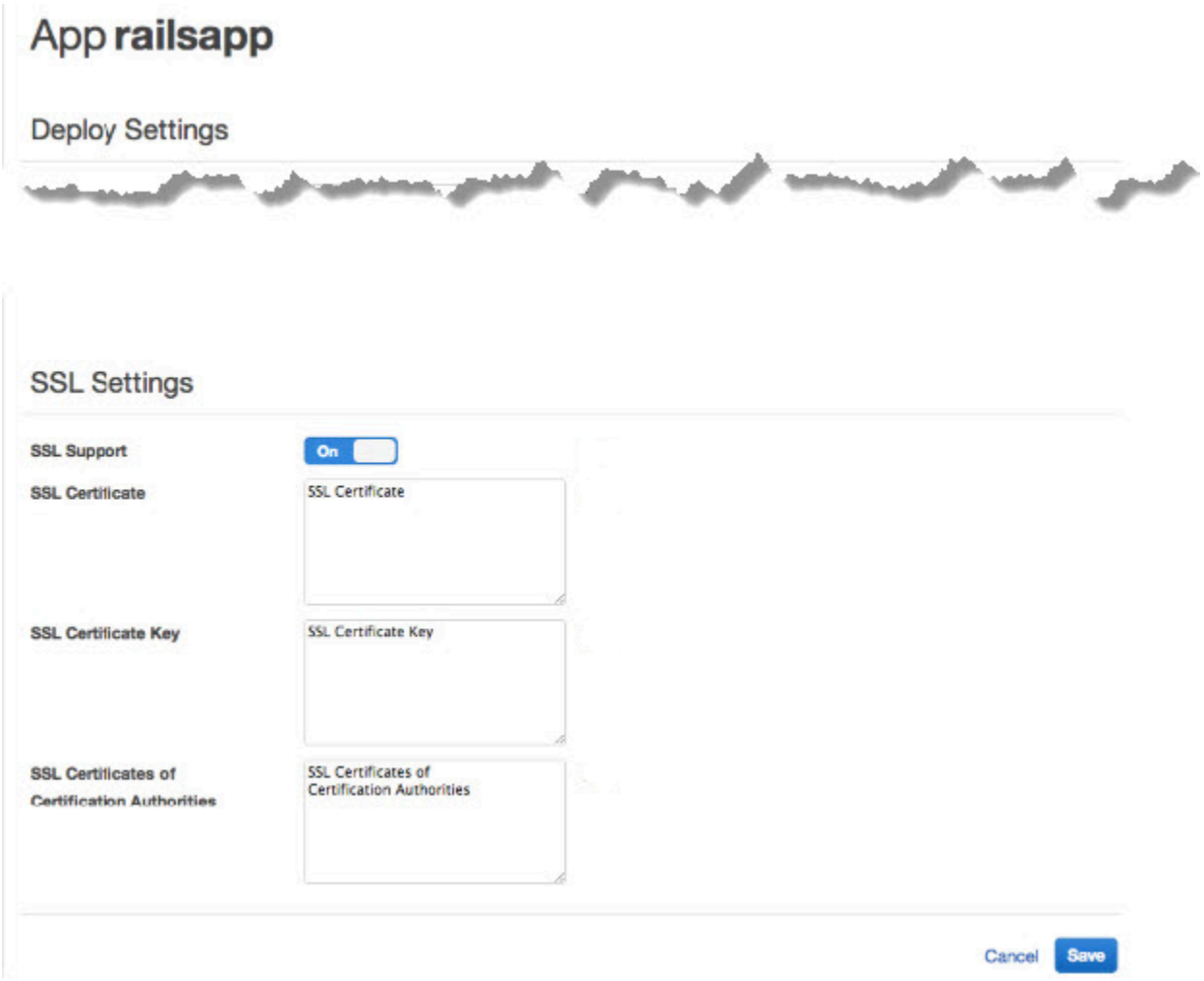
Copiare i contenuti del file della chiave privata (file .pem) nella casella. Deve avere un aspetto simile al seguente:

```
----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQC0CYklJY5r4vV2NHQYEpwtsLuMMBhylMrgBShKq+HHVLYQQCL6
+wGIiRq5qXqZlRXje3GM5Jvcm6q0R71MfRIl1FuzKyqDtneZaAIEYniZibHiUnmO
/UNqpFDosw/6hY3ONk0fSBlU4ivD0Gjpf6J80jL3DJ4R23Ed0sdL4pRT3QIDAQAB
AoGBAKmMfWrNRqYVtGKgnWB6Tji9QrKQLMXjmHeGg95mppdJELiXHhpMvrHtpIyK
...
-----END RSA PRIVATE KEY-----
```

SSL certificates of Certification Authorities (Certificati SSL delle Certification Authority)

Se si dispone di un file della catena del certificato, incollare il contenuto nella casella.

> **Note**
>
> Se si sta usando Nginx, è necessario lasciare la casella vuota. Se si dispone di un file della catena del certificato, aggiungerla al file del certificato della chiave pubblica in SSL Certificate (Certificato SSL).



Dopo aver fatto clic su Save (Salva), ridistribuire l'applicazione per aggiornare le proprie istanze online.

Per i livelli di application server integrati, AWS OpsWorks Stacks aggiorna automaticamente la configurazione del server. Al termine della distribuzione, è possibile verificare che l'installazione OpenSSL è andata a buon fine, come segue.

Per verificare un'installazione OpenSSL

1. Andare alla pagina Instances (Istanze).
2. Eseguire l'app facendo clic sull'indirizzo IP delle istanze del server di applicazione o, se si sta utilizzando un sistema di bilanciamento del carico, l'indirizzo IP del sistema di bilanciamento del carico.
3. Modificare il prefisso dell'indirizzo IP da **`http://`** a **`https://`** e aggiornare il browser per verificare che la pagina venga caricata correttamente con SSL.

Gli utenti che hanno configurato le app per l'esecuzione in Mozilla Firefox talvolta ottengono il seguente errore di certificato: `SEC_ERROR_UNKNOWN_ISSUER`. Questo errore può essere causato dalla funzionalità di sostituzione dei certificati nei programmi antivirus e antimalware dell'organizzazione, da alcuni tipi di software di monitoraggio e filtro del traffico di rete o da malware. Per ulteriori informazioni su come risolvere questo errore, consulta Come interpretare i codici di errore e risolvere i problemi di sicurezza per i siti web sicuri sul sito Web di supporto di Mozilla Firefox.

Per tutti gli altri livelli, inclusi i livelli personalizzati, AWS OpsWorks Stacks aggiunge semplicemente le impostazioni SSL agli attributi `deploy` dell'app. È necessario implementare una ricetta personalizzata per recuperare le informazioni dall'oggetto nodo e configurare il server in modo appropriato.

Libri di ricette e ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks utilizza i libri di cucina Chef per gestire attività come l'installazione e la configurazione di pacchetti e la distribuzione di app. Questa sezione descrive come utilizzare i libri di cucina con AWS OpsWorks Stacks. Per ulteriori informazioni, consulta Chef.

> **Note**
>
> AWS OpsWorks Stacks attualmente supporta le versioni 12, 11.10.4, 11.4.4 e 0.9.15.5 di Chef. Tuttavia, Chef 0.9.15.5 è obsoleto e non è consigliabile utilizzarlo per i nuovi stack. Per comodità, agli stack viene in genere fatto riferimento mediante i rispettivi numeri di versione principale e secondaria. Gli stack che eseguono Chef 0.9 o 11.4 utilizzano Chef Solo, mentre gli stack che eseguono Chef 12 o 11.10 utilizzano Chef Client in modalità locale. Per gli stack Linux, puoi utilizzare la funzionalità Configuration Manager (Gestore configurazione) per specificare la versione di Chef da utilizzare quando crei uno stack. Gli stack di Windows devono utilizzare Chef 12.2. Per ulteriori informazioni, incluse le linee guida per la migrazione degli stack in versioni più recenti di Chef, consulta Versione di Chef.

Argomenti

- Repository dei libri di ricette
- Versione di Chef
- Versioni di Ruby
- Installazione di libri di ricette personalizzati
- Aggiornamento dei libri di ricette personalizzati
- Esecuzione di ricette

Repository dei libri di ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

I tuoi libri di ricette personalizzati devono essere archiviati in un repository online: un archivio come un file .zip o un gestore di controllo del codice sorgente come Git. Uno stack può avere un solo repository di libri di ricette personalizzati, ma tale repository può contenere un numero indefinito

di libri di ricette. Quando installi o aggiorni i libri di cucina, AWS OpsWorks Stacks installa l'intero repository in una cache locale su ciascuna istanza dello stack. Quando necessita, ad esempio, di eseguire una o più ricette, un'istanza utilizza il codice della cache locale.

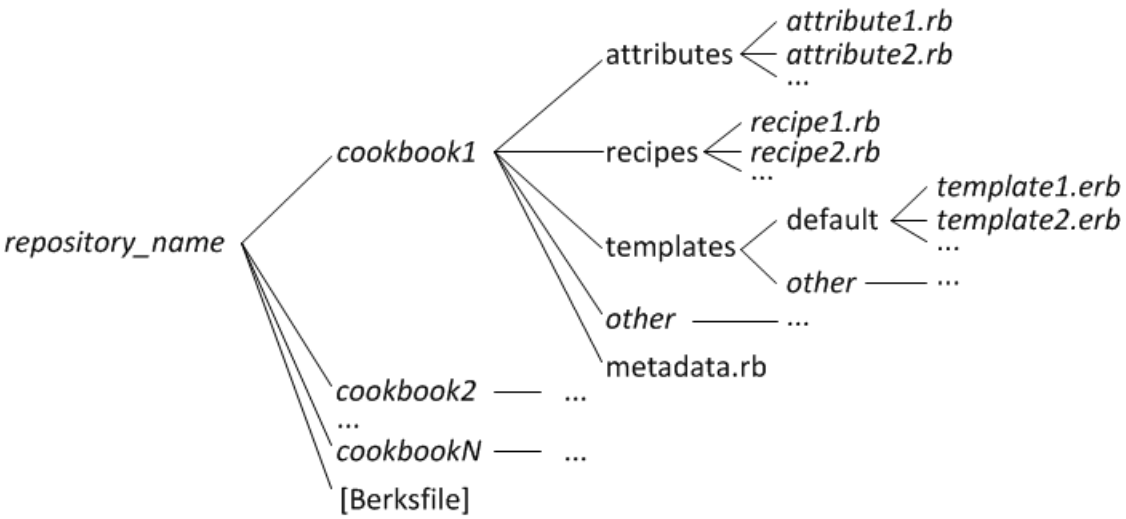
Di seguito viene descritto come strutturare il repository del libro di ricette in base al tipo. Il testo in corsivo nelle illustrazioni rappresenta i nomi di directory e di file definiti dall'utente, incluso il nome del repository o dell'archivio.

Gestore di controllo del codice sorgente

AWS OpsWorks Stacks supporta i seguenti gestori del controllo del codice sorgente:

- Stack Linux: Git e Subversion
- Windows stack — Git

Di seguito viene mostrata la struttura di file e directory richiesta:



- Le directory di libri di ricette devono essere tutte al primo livello.

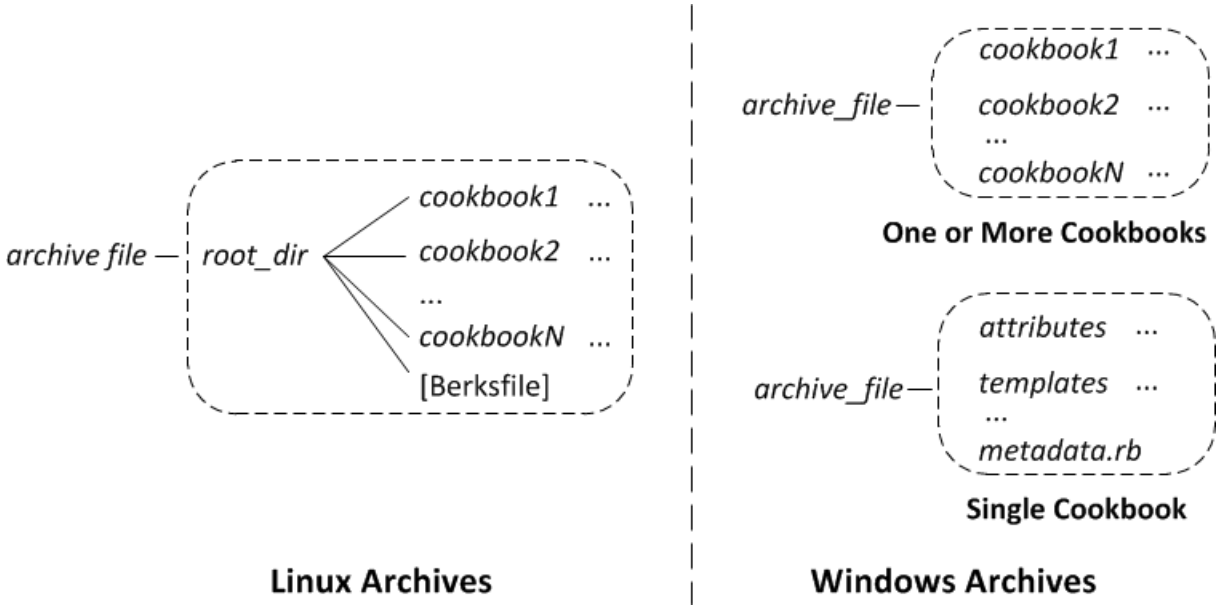
Archive (Archivia)

AWS OpsWorks Stacks supporta i seguenti archivi:

- Stack Linux: file zip, gzip, bzip2 o tarball, archiviati su Amazon S3 o su un sito Web (archivio HTTP).

  AWS OpsWorks Stacks non supporta archivi tar non compressi.
- Stack di Windows: file zip e tgz (tar compresso con gzip), archiviati su Amazon S3.

Di seguito viene mostrata la struttura di file e directory richiesta, che varia a seconda che si esegua uno stack Linux o Windows. La struttura del libro di ricette è la stessa dei repository SCM, perciò è rappresentata da un'ellissi (...).



- Pile Linux: le directory dei libri di cucina devono essere contenute in una directory principale.
- Pile di Windows: i libri di cucina devono trovarsi al livello più alto dell'archivio.

  Se disponi di un solo libro di ricette, puoi omettere la relativa directory e inserire i file del libro di ricette nel primo livello. In questo caso, AWS OpsWorks Stacks ottiene il nome del libro di ricette da metadata.rb.

Ogni directory del libro di ricette include almeno uno, ma generalmente tutti, i seguenti file e directory standard, che devono utilizzare nomi standard:

- `attributes`— I file degli attributi del libro di cucina.
- `recipes`— I file di ricette del ricettario.
- `templates`— I file modello del ricettario.
- *`other`*— Directory opzionali definite dall'utente che contengono altri tipi di file, come definizioni o specifiche.
- `metadata.rb`— I metadati del ricettario.

  Per Chef 11.10 e versioni successive, se le tue ricette dipendono da altri libri di ricette, devi includere le istruzioni `depends` corrispondenti nel file `metadata.rb` del tuo libro di ricette. Ad esempio, se il tuo libro di ricette include una ricetta con un'istruzione come `include_recipe`

`anothercookbook::somerecipe`, il file `metadata.rb` del libro di ricette include la riga seguente: `depends "anothercookbook"`. Per ulteriori informazioni, consulta l'articolo sulle informazioni sui metadati dei libri di ricette.

I modelli devono essere in una sottodirectory della directory `templates`, che contiene almeno una e, facoltativamente, più sottodirectory. Tali sottodirectory possono a loro volta avere altre sottodirectory.

- Solitamente i modelli hanno una sottodirectory `default`, che contiene i file modello che Chef utilizza per impostazione predefinita.
- other rappresenta sottodirectory facoltative che possono essere utilizzate per modelli specifici del sistema operativo.
- Chef utilizza automaticamente il modello dalla sottodirectory appropriata, in base alle convenzione di denominazione descritte in Specificità dei file. Ad esempio, per i sistemi operativi Amazon Linux e Ubuntu, puoi inserire modelli specifici del sistema operativo in sottodirectory denominate, rispettivamente, `amazon` o `ubuntu`.

I dettagli in merito a come gestire libri di ricette personalizzati variano in base al tipo di repository preferito.

Per utilizzare un archivio

1. Implementare i libri di ricette utilizzando la struttura di cartella visualizzata nella sezione precedente.
2. Crea un archivio compresso e caricalo in un bucket Amazon S3 o un sito Web.

   Se si aggiornano i libri di ricette, è necessario creare e caricare un nuovo file di archivio. I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Per utilizzare un SCM

1. Configurare un repository Git o Subversion utilizzando la struttura mostrata in precedenza.
2. Se lo si desidera, è possibile utilizzare le funzionalità di controllo della versione del repository per implementare più rami o versioni.

Se aggiorni i tuoi libri di cucina, puoi farlo in una nuova filiale e OpsWorks utilizzare direttamente la nuova versione. È anche possibile specificare particolari versioni con tag. Per informazioni dettagliate, consultare Specifica di un archivio di libri di ricette personalizzati.

Installazione di libri di ricette personalizzatidescrive come fare in modo che AWS OpsWorks Stacks installi il tuo repository di libri di cucina sulle istanze dello stack.

> **Important**
>
> Dopo aver aggiornato i libri di cucina esistenti nel repository, devi eseguire il comando `update_cookbooks` stack per indirizzare AWS OpsWorks Stacks ad aggiornare la cache locale di ogni istanza online. Per ulteriori informazioni, consulta Esecuzione dei comandi dello stack.

Versione di Chef

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks supporta più versioni di Chef. La versione viene selezionata quando si crea lo stack. AWS OpsWorks Stacks installa quindi quella versione di Chef su tutte le istanze dello stack insieme a una serie di ricette integrate compatibili con quella versione. Se installi ricette personalizzate, devono essere compatibile con la versione di Chef dello stack.

AWS OpsWorks Stacks attualmente supporta le versioni Chef 12, 11.10, 11.4 e 0.9 per gli stack Linux e Chef 12.2 (attualmente Chef 12.22) per gli stack Windows. Per comodità, agli stack viene in genere fatto riferimento mediante i rispettivi numeri di versione principale e secondaria. Per gli stack Linux, puoi utilizzare la funzionalità Configuration Manager (Gestore configurazione) per specificare la versione di Chef da utilizzare quando crei uno stack. Gli stack di Windows devono utilizzare Chef

12.2. Per ulteriori informazioni, incluse le linee guida per la migrazione degli stack in versioni più recenti di Chef, consulta Versione di Chef. Per tutte le informazioni sulle versioni, consulta AWS OpsWorks Sistemi operativi impilati.

Chef 12.2

Il supporto per Chef 12.2 è stato introdotto a maggio 2015 e viene utilizzato solo dagli stack di Windows. La versione corrente di Chef per gli stack Windows è Chef 12.22. Viene eseguito con Chef 2.3.6 e utilizza chef-client nella modalità locale, che consente di avviare un server Chef in-memory locale chiamato chef-zero. La presenza di questo server permette alle ricette di utilizzare i contenitori di dati e la ricerca di Chef. Il supporto è caratterizzato da alcune limitazioni, che vengono descritte in Implementazione delle ricette: Chef 12.2, ma puoi eseguire numerosi libri di ricette della community senza alcuna modifica.

Chef 12

Il supporto di Chef 12 è stato introdotto nel dicembre 2015 e viene utilizzato solo dagli stack Linux. Viene eseguito con Ruby 2.1.6 o 2.2.3 e usa chef-client in modalità locale, che consente alle ricette di utilizzare i contenitori di dati e la ricerca di Chef. Per ulteriori informazioni, consulta AWS OpsWorks Sistemi operativi impilati.

Chef 11.10

Il supporto di Chef 11.10 è stato introdotto nel marzo 2014 e viene utilizzato solo dagli stack Linux. Viene eseguito con Ruby 2.0.0 e usa chef-client in modalità locale, che consente alle ricette di utilizzare i contenitori di dati e la ricerca di Chef. Il supporto è caratterizzato da alcune limitazioni, che vengono descritte in Implementazione delle ricette: Chef 11.10, ma puoi eseguire numerosi libri di ricette della community senza alcuna modifica. Per gestire le dipendenze dei libri di ricette, puoi utilizzare anche Berkshelf. Le versioni di Berkshelf supportate dipendono dal sistema operativo. Per ulteriori informazioni, consulta AWS OpsWorks Sistemi operativi impilati. Non puoi creare stack CentOS che utilizzano Chef 11.10.

Chef 11.4

Il supporto di Chef 11.4 è stato introdotto nel luglio 2013 e viene utilizzato solo dagli stack Linux. Viene eseguito con Ruby 1.8.7 e usa chef-solo, che non supporta i contenitori di dati e la ricerca di Chef. Spesso puoi usare libri di cucina della community che dipendono da tali funzionalità con AWS OpsWorks Stacks, ma devi modificarli come descritto in. Migrazione in una nuova versione di Chef Non puoi creare stack CentOS che utilizzano Chef 11.4. Gli stack Chef 11.4 non sono supportati negli endpoint regionali al di fuori della regione Stati Uniti orientali (Virginia settentrionale).

Chef 0.9

Chef 0.9 viene utilizzato solo dagli stack Linux e non è più supportato. Si noti quanto segue:

- Non puoi utilizzare la console per creare un nuovo stack Chef 0.9.

  Devi utilizzare la CLI o l'API oppure devi creare uno stack con una versione diversa di Chef e quindi modificare la configurazione dello stack.
- Le nuove funzionalità AWS OpsWorks Stacks non sono disponibili per gli stack Chef 0.9.
- Le nuove versioni dei sistemi operativi forniranno solo un supporto limitato per gli stack Chef 0.9.

  In particolare, Amazon Linux 2014.09 e versioni successive non supportano gli stack Chef 0.9 con livelli Rails App Server che dipendono da Ruby 1.8.7.
- Le nuove regioni AWS, inclusa l'Europa (Francoforte), non supportano gli stack Chef 0.9.

> **Note**
>
> Non consigliamo di utilizzare Chef 0.9 per i nuovi stack. È consigliabile eseguire la migrazione degli stack esistenti alla versione più aggiornata di Chef il prima possibile.

Se desideri utilizzare libri di cucina della community con AWS OpsWorks Stacks, ti consigliamo di specificare Chef 12 per i nuovi stack Linux e di migrare gli stack Linux esistenti su Chef 12. Puoi utilizzare la console, l'API o la CLI di AWS OpsWorks Stacks per migrare gli stack esistenti a una versione più recente di Chef. Per ulteriori informazioni, consulta Migrazione in una nuova versione di Chef.

Argomenti

- Implementazione di ricette per Chef 12.2 Stacks
- Implementazione delle ricette per gli stack Chef 12
- Implementazione delle ricette per gli stack Chef 11.10
- Implementazione delle ricette per gli stack Chef 11.4
- Migrazione di un stack Linux esistente in una nuova versione di Chef

Implementazione di ricette per Chef 12.2 Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Chef 12.2 (attualmente Chef 12.22) è disponibile solo per gli stack Windows, che devono eseguire tale versione di Chef.

- In alcuni casi le ricette devono utilizzare gli attributi e le risorse specifici di Windows.

  Per ulteriori informazioni, consulta l'argomento relativo a Chef per Microsoft Windows.
- Le esecuzioni di Chef utilizzano Ruby 2.3.6. Pertanto, le ricette possono utilizzare la nuova sintassi Ruby.
- Le ricette possono utilizzare i contenitori di dati e la ricerca di Chef.

  Chef 12.2 stacks può utilizzare molti libri di cucina della community senza modifiche. Per ulteriori informazioni, consultare Utilizzo della ricerca di Chef e Utilizzo di contenitori di dati.
- La maggior parte degli attributi di configurazione e distribuzione degli stack descritti in AWS OpsWorks Riferimento a Stacks Data Bag e Attributi dei libri di ricette predefiniti è disponibile per le ricette Windows.

  Puoi utilizzare la ricerca di Chef per recuperare tali valori degli attributi. Per vedere un esempio, consulta Ottenimento dei valori degli attributi con la ricerca di Chef. Per un elenco degli attributi, consulta AWS OpsWorks Riferimento a Stacks Data Bag.

Implementazione delle ricette per gli stack Chef 12

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Gli stack Chef 12 offrono i seguenti vantaggi rispetto agli stack Chef 11.10:

- Le esecuzioni di Chef utilizzano Ruby 2.1.6. Pertanto, le ricette possono utilizzare la nuova sintassi Ruby.
- Gli stack Chef 12 possono utilizzare un numero maggiore di libri di ricette della community, senza alcuna modifica. Senza i libri di ricette predefiniti, non si verificheranno più conflitti a livello di nomi tra i libri di ricette predefiniti e quelli personalizzati.
- Non sei più limitato alle versioni di Berkshelf per le quali AWS OpsWorks Stacks ha fornito pacchetti predefiniti. Berkshelf non è più installato sulle AWS OpsWorks istanze Stacks in Chef 12. Puoi invece utilizzare qualsiasi versione di Berkshelf sulla workstation locale.
- Ora esiste una netta separazione tra i libri di cucina integrati forniti da AWS OpsWorks Stacks con Chef 12 (Elastic Load Balancing, Amazon RDS e Amazon ECS) e i libri di cucina personalizzati. Ciò semplifica la risoluzione dei problemi relativi alle esecuzioni di Chef non riuscite.

Implementazione delle ricette per gli stack Chef 11.10

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Gli stack Chef 11.10 offrono i seguenti vantaggi rispetto agli stack Chef 11.4:

- Le esecuzioni di Chef utilizzano Ruby 2.0.0. Pertanto, le ricette possono utilizzare la nuova sintassi Ruby.
- Le ricette possono utilizzare i contenitori di dati e la ricerca di Chef.

  Gli stack Chef 11.10 possono utilizzare numerosi libri di ricette della community, senza alcuna modifica.

- Puoi utilizzare Berkshelf per la gestione dei libri di ricette.

  Berkshelf offre maggiore flessibilità per quanto riguarda la gestione dei libri di ricette personalizzati e l'utilizzo dei libri di ricette della community in uno stack.
- I libri di ricette devono dichiarare le dipendenze `metadata.rb`.

  Se il tuo libro di ricette dipende da un altro libro di ricette, devi includere tale dipendenza nel file `metadata.rb` del tuo libro di ricette. Ad esempio, se il tuo libro di ricette include una ricetta con un'istruzione come `include_recipe anothercookbook::somerecipe`, il file `metadata.rb` del libro di ricette include la riga seguente: `depends "anothercookbook"`.
- AWS OpsWorks Stacks installa un client MySQL sulle istanze di uno stack solo se lo stack include un livello MySQL.
- AWS OpsWorks Stacks installa un client Ganglia sulle istanze di uno stack solo se lo stack include un livello Ganglia.
- Se una distribuzione esegue `bundle install` e l'installazione ha esito negativo, anche la distribuzione avrà esito negativo.

> **Important**
>
> Non riutilizzare i nomi di libri di ricette predefiniti per i libri di ricette personalizzati e per quelli della community. I libri di ricette personalizzati con lo stesso nome dei libri di ricette predefiniti potrebbero avere esito negativo. Per un elenco completo dei libri di cucina integrati disponibili con gli stack Chef 11.10, 11.4 e 0.9, consulta il repository opsworks-cookbooks su. GitHub
> I libri di ricette con caratteri non ASCII eseguiti correttamente sugli stack Chef 0.9 e 11.4 potrebbero avere esito negativo su uno stack Chef 11.10. Il motivo è che gli stack Chef 11.10 usano Ruby 2.0.0 per l'esecuzione di Chef, che prevede una codifica molto più rigida rispetto a Ruby 1.8.7. Per garantire che tali libri di ricette vengano eseguiti correttamente sugli stack Chef 11.10, ogni file che usa caratteri non ASCII deve avere un commento nella parte superiore che fornisce un suggerimento sulla codifica. Ad esempio, per la codifica UTF-8, il commento potrebbe essere `# encoding: UTF-8`. Per ulteriori informazioni sulla codifica di Ruby 2.0.0, consulta l'argomento relativo alla codifica di Ruby 2.0.0.

Argomenti

- Installazione e precedenza dei libri di ricette
- Utilizzo della ricerca di Chef

- Utilizzo di contenitori di dati
- Utilizzo di Berkshelf

Installazione e precedenza dei libri di ricette

La procedura per installare i libri di cucina AWS OpsWorks Stacks funziona in modo leggermente diverso per gli stack di Chef 11.10 rispetto alle versioni precedenti di Chef. Per gli stack Chef 11.10, dopo che AWS OpsWorks Stacks ha installato i ricettari integrati, personalizzati e Berkshelf, li unisce in una directory comune nell'ordine seguente:

1. Libri di ricette predefiniti.
2. Libri di ricette di Berkshelf, se presenti.
3. Libri di ricette personalizzati, se presenti.

Quando AWS OpsWorks Stacks esegue questa unione, copia l'intero contenuto delle directory, comprese le ricette. Se sono presenti duplicati, vengono applicate le regole seguenti:

- I contenuti dei libri di ricette di Berkshelf hanno la precedenza sui libri di ricette predefiniti.
- I contenuti dei libri di ricette personalizzati hanno la precedenza sui libri di ricette di Berkshelf.

Per illustrare il funzionamento di questo processo, prendiamo in considerazione il seguente scenario, dove le tre directory di libri di ricette includono tutte un libro di ricette denominato `mycookbook`:

- Libri di cucina integrati: `mycookbook` include un file di attributi denominato`someattributes.rb`, un file modello denominato e una ricetta `sometemplate.erb` denominata. `somerecipe.rb`
- Libri di cucina Berkshelf: include e. `mycookbook` `sometemplate.erb` `somerecipe.rb`
- Libri di cucina personalizzati: include. `mycookbook` `somerecipe.rb`

Il libro di ricette unito contiene i seguenti elementi:

- `someattributes.rb` dal libro di ricette predefinito.
- `sometemplate.erb` dal libro di ricette di Berkshelf.
- `somerecipe.rb` dal libro di ricette personalizzato.

> **Important**
>
> Non devi personalizzare lo stack Chef 11.10 mediante la copia dell'intero libro di ricette predefinito nel tuo archivio e quindi mediante la modifica di alcune sue parti. In questo modo, sostituisci l'intero libro di ricette predefinito, comprese le ricette. Se AWS OpsWorks Stacks aggiorna quel ricettario, il tuo stack non trarrà vantaggio da tali aggiornamenti a meno che non aggiorni manualmente la tua copia privata. Per ulteriori informazioni su come personalizzare gli stack, consulta Personalizzazione degli stack AWS OpsWorks.

Utilizzo della ricerca di Chef

Puoi utilizzare il metodo `search` di Chef nelle ricette per eseguire query sui dati dello stack. Usi la stessa sintassi del server Chef, ma AWS OpsWorks Stacks ottiene i dati dall'oggetto nodo locale invece di interrogare un server Chef. Questi dati includono:

- Gli attributi di configurazione e distribuzione dello stack dell'istanza.
- Gli attributi ricavati dai file di attributi dei libri di ricette predefiniti e personalizzati dell'istanza.
- Dati di sistema raccolti da Ohai.

Gli attributi di configurazione e distribuzione dello stack contengono la maggior parte delle informazioni che le ricette in genere ottengono tramite la ricerca, inclusi dati come nomi host e indirizzi IP per ogni istanza online dello stack. AWS OpsWorks Stacks aggiorna questi attributi per ogni evento del ciclo di vita, garantendo che riflettano accuratamente lo stato attuale dello stack. Ciò significa che spesso nello stack puoi utilizzare le ricette della community che dipendono dalla funzionalità di ricerca, senza alcuna modifica. Il metodo di ricerca restituisce i dati appropriati, derivati dagli attributi di configurazione e distribuzione dello stack, anziché da un server.

La limitazione principale di AWS OpsWorks Stacks search è che gestisce solo i dati nell'oggetto del nodo locale, in particolare la configurazione dello stack e gli attributi di distribuzione. Per questo motivo, i seguenti tipi di dati potrebbero non essere disponibili tramite la ricerca:

- Attributi definiti localmente su altre istanze.

  Se una ricetta definisce un attributo localmente, tali informazioni non vengono riportate al servizio AWS OpsWorks Stacks, quindi non è possibile accedere a tali dati da altre istanze utilizzando la ricerca.
- Attributi `deploy` personalizzati

Puoi specificare dati JSON personalizzati quando distribuisci un'app; gli attributi corrispondenti vengono installati nelle istanze dello stack per tale distribuzione. Tuttavia, se esegui la distribuzione solo sulle istanze selezionate, gli attributi vengono installati solo su tali istanze. Le query relative agli attributi JSON personalizzati avranno esito negativo su tutte le altre istanze. Inoltre, gli attributi personalizzati vengono inclusi nei dati JSON di configurazione e distribuzione dello stack solo per la distribuzione specifica. Tali dati sono accessibili solo finché il successivo evento del ciclo di vita non installa un nuovo set di attributi di configurazione e distribuzione dello stack. Si noti che se specifichi dati JSON personalizzati per lo stack, gli attributi vengono installati su ogni istanza per ogni evento del ciclo di vita, risultando così sempre accessibili tramite la ricerca.

- Dati Ohai da altre istanze.

  Lo strumento Ohai di Chef recupera una serie di dati di sistema su un'istanza e aggiunge tali dati all'oggetto nodo. Questi dati vengono archiviati in locale e non vengono restituiti al servizio AWS OpsWorks Stacks. Pertanto, la ricerca non è in grado di accedere ai dati Ohai da altre istanze. Tuttavia, alcuni di questi dati potrebbero essere inclusi negli attributi di configurazione e distribuzione dello stack.

- Istanze offline.

  Gli attributi di configurazione e distribuzione dello stack contengono i dati solo per le istanze online.

Il seguente segmento di ricetta mostra come recuperare l'indirizzo IP privato di un'istanza del livello PHP mediante la ricerca.

```
appserver = search(:node, "role:php-app").first
Chef::Log.info("The private IP is '#{appserver[:private_ip]}'")
```

> **Note**
>
> Quando AWS OpsWorks Stacks aggiunge gli attributi di configurazione e distribuzione dello stack all'oggetto nodo, crea effettivamente due set di attributi di livello, ciascuno con gli stessi dati. Un set si trova nel `layers` namespace, che è il modo in cui AWS OpsWorks Stacks archivia i dati. L'altro set si trova nello spazio dei nomi `role` namespace, che è il modo in cui il server Chef archivia i dati equivalenti. Lo scopo del `role` namespace è consentire l'esecuzione del codice di ricerca implementato per il server Chef su un'istanza Stacks. AWS OpsWorks Se stai scrivendo codice specifico per AWS OpsWorks Stacks, puoi usare uno

dei due `layers:php-app` o `role:php-app` l'esempio precedente e `search` restituiresti lo stesso risultato.

Utilizzo di contenitori di dati

Puoi utilizzare il metodo `data_bag_item` di Chef nelle ricette per eseguire query sulle informazioni incluse nei contenitori di dati. Puoi utilizzare la stessa sintassi valida per il server Chef. Tuttavia, AWS OpsWorks Stacks recupera i dati dagli attributi di configurazione e distribuzione dello stack dell'istanza. Tuttavia, AWS OpsWorks Stacks attualmente non supporta gli ambienti Chef, quindi `node.chef_environment` ritorna sempre. `_default`

Puoi creare un contenitore di dati utilizzando dati JSON personalizzati per aggiungere uno o più attributi all'attributo `[:opsworks][:data_bags]`. L'esempio seguente mostra il formato generale per la creazione di un contenitori di dati in formato JSON personalizzato.

> **Note**
>
> Non puoi creare un contenitore di dati mediante la sua aggiunta nell'archivio di libri di ricette. Devi utilizzare dati JSON personalizzati.

```
{
  "opsworks": {
    "data_bags": {
      "bag_name1": {
        "item_name1: {
          "key1" : “value1”,
          "key2" : “value2”,
          ...
        }
      },
      "bag_name2": {
        "item_name1": {
          "key1" : “value1”,
          "key2" : “value2”,
          ...
        }
      },
      ...
```

```
    }
  }
}
```

In genere specifichi dati JSON personalizzati per lo stack. Questa operazione installa gli attributi personalizzati in ogni istanza per ciascun evento del ciclo di vita successivo. Puoi anche specificare dati JSON personalizzati quando distribuisci un'app, ma tali attributi vengono installati solo per la distribuzione specifica e potrebbero venire installati solo in un set selezionato di istanze. Per ulteriori informazioni, consulta Distribuzione di app.

Il seguente esempio di dati JSON personalizzati crea un contenitore di dati denominato `myapp`. Include un elemento, `mysql`, con due coppie chiave-valore.

```
{ "opsworks": {
    "data_bags": {
      "myapp": {
        "mysql": {
          "username": "default-user",
          "password": "default-pass"
        }
      }
    }
  }
}
```

Per utilizzare i dati nella ricetta, puoi eseguire una chiamata a `data_bag_item` e passare i nomi del contenitore di dati e dei valori, come illustrato nel seguente segmento.

```
mything = data_bag_item("myapp", "mysql")
Chef::Log.info("The username is '#{mything['username']}' ")
```

Per modificare i dati nel contenitore di dati, è sufficiente modificare i dati JSON personalizzati. Questa operazione li installerà nelle istanze dello stack per il successivo evento del ciclo di vita.

Utilizzo di Berkshelf

Con gli stack Chef 0.9 e Chef 11.4 puoi installare solo un archivio di libri di ricette personalizzati. Con gli stack Chef 11.10 puoi utilizzare Berkshelf per gestire i libri di ricette e le relative dipendenze.

Berkshelf ti consente inoltre di installare libri di ricette da più archivi. Per ulteriori informazioni, consulta Creazione di pacchetti di dipendenze di libri di ricette a livello locale. In particolare, con Berkshelf, puoi installare libri di cucina della community AWS OpsWorks compatibili con Stacks direttamente dai loro repository invece di doverli copiare nel tuo repository di libri di cucina personalizzato. Le versioni di Berkshelf supportate dipendono dal sistema operativo. Per ulteriori informazioni, consulta AWS OpsWorks Sistemi operativi impilati.

Per usare Berkshelf, devi abilitarlo in modo esplicito, come descritto in Installazione di libri di ricette personalizzati. Devi quindi includere un file `Berksfile` nella directory principale dell'archivio di libri di ricette, in cui specifichi i libri di ricette da installare.

Per specificare un'origine esterna di libri di ricette in un file Berksfile, all'inizio del file devi includere un attributo di origine che specifica l'URL dell'archivio predefinito. Berkshelf cercherà i libri di cucina nel codice sorgente a meno che tu non specifichi esplicitamente un repository. URLs Devi quindi includere una riga per ogni libro di ricette che vuoi installare usando il formato seguente:

```
cookbook 'cookbook_name', ['>= cookbook_version'], [cookbook_options]
```

I campi dopo `cookbook` indicano il libro di ricette specifico.

- *`cookbook_name`*— (Obbligatorio) Specificate il nome del libro di cucina.

  Se non includi altri campi, Berkshelf installa il ricettario dalla fonte specificata. URLs
- *`cookbook_version`*— (Facoltativo) Specificate la versione o le versioni del libro di cucina.

  Puoi utilizzare un prefisso, ad esempio = o >=, per specificare una determinata versione o un intervallo di versioni accettabili. Se non specifichi una versione, Berkshelf installa la più versione recente.
- *`cookbook_options`*— (Facoltativo) L'ultimo campo è un hash contenente una o più coppie chiave-valore che specificano opzioni come la posizione del repository.

  Ad esempio, puoi includere una chiave `git` per designare un determinato archivio Git e una chiave `tag` per designare un determinato ramo dell'archivio. Il metodo migliore per essere sicuro di installare il libro di ricette desiderato è specificare il ramo dell'archivio.

> **Important**
>
> Non dichiarare i libri di ricette includendo una riga `metadata` nel file Berksfile e dichiarando le dipendenze del libro di ricette in `metadata.rb`. Per garantirne il corretto funzionamento, entrambi i file devono trovarsi nella stessa directory. Con AWS OpsWorks Stacks, il Berksfile deve trovarsi nella directory principale del repository, ma i file devono trovarsi nelle rispettive directory `metadata.rb` dei libri di cucina. È consigliabile invece dichiarare in modo esplicito i libri di ricette esterni nel file Berksfile.

Di seguito è riportato un esempio di file Berksfile che mostra diversi modi per specificare i libri di ricette. Per ulteriori informazioni su come creare un file Berksfile, consulta Berkshelf.

```
source "https://supermarket.chef.io"

cookbook 'apt'
cookbook 'bluepill', '>= 2.3.1'
cookbook 'ark', git: 'git://github.com/opscode-cookbooks/ark.git'
cookbook 'build-essential', '>= 1.4.2', git: 'git://github.com/opscode-cookbooks/build-
essential.git', tag: 'v1.4.2'
```

Questo file installa i seguenti di libri di ricette:

- La versione più recente di `apt` dall'archivio di libri di ricette della community.
- La versione più recente di `bluepill` dai libri di ricette della community, purché la relativa versione sia 2.3.1 o successive.
- La versione più recente di `ark` da un archivio specificato.

  L'URL di questo esempio si riferisce a un archivio pubblico di libri di cucina della comunità su GitHub, ma puoi installare libri di cucina da altri repository, inclusi gli archivi privati. Per ulteriori informazioni, consulta Berkshelf.
- Il libro di ricette `build-essential` dal ramo v1.4.2 dell'archivio specificato.

Un archivio di libri di ricette personalizzati può contenere libri di ricette personalizzati e un file Berksfile. In tal caso, AWS OpsWorks Stacks installa entrambi i set di libri di cucina, il che significa che un'istanza può avere fino a tre repository di libri di cucina.

- I libri di ricette predefiniti vengono installati in `/opt/aws/opsworks/current/cookbooks`.
- Se l'archivio di libri di ricette personalizzati contiene libri di ricette, essi vengono installati in `/opt/aws/opsworks/current/site-cookbooks`.
- Se hai abilitato Berkshelf e l'archivio di libri di ricette personalizzati include un file Berksfile, i libri di ricette specificati vengono installati in `/opt/aws/opsworks/current/berkshelf-cookbooks`.

I ricettari integrati e i ricettari personalizzati vengono installati su ogni istanza durante la configurazione e non vengono successivamente aggiornati a meno che non si esegua manualmente il comando Update Custom Cookbooks stack. AWS OpsWorks Stacks viene eseguito `berks install` per ogni esecuzione di Chef, quindi i tuoi libri di cucina Berkshelf vengono aggiornati per ogni evento del ciclo di vita, secondo le seguenti regole:

- Se l'archivio include una nuova versione del libro di ricette, questa operazione aggiorna il libro di ricette dall'archivio.
- In caso contrario, questa operazione aggiorna i libri di ricette di Berkshelf da una cache locale.

> **Note**
>
> L'operazione sovrascrive i libri di ricette di Berkshelf. Pertanto, se hai modificato le copie locali dei libri di ricette, le modifiche verranno sovrascritte. Per ulteriori informazioni, consulta Berkshelf.

Puoi anche aggiornare i libri di ricette di Berkshelf eseguendo il comando Update Custom Cookbooks (Aggiorna libri di ricette personalizzati) a livello di stack. Questa operazione aggiorna sia i libri di ricette di Berkshelf che quelli personalizzati.

Implementazione delle ricette per gli stack Chef 11.4

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Important**
>
> Non riutilizzare i nomi di libri di ricette predefiniti per i libri di ricette personalizzati e per quelli della community. I libri di ricette personalizzati con lo stesso nome dei libri di ricette predefiniti potrebbero avere esito negativo. Per un elenco completo dei libri di cucina integrati disponibili con gli stack Chef 11.10, 11.4 e 0.9, consulta il repository opsworks-cookbooks su. GitHub

Il limite principale degli stack Chef 11.4 è data dal fatto che le ricette non possono utilizzare i contenitori di dati e la ricerca di Chef. Tuttavia, AWS OpsWorks Stacks installa gli attributi di configurazione e distribuzione dello stack su ogni istanza che contengono molte delle informazioni che si ottengono con la ricerca, tra cui:

- Dati definiti dall'utente dalla console, ad esempio nomi dell'host o delle app.
- Dati di configurazione dello stack generati dal servizio AWS OpsWorks Stacks, come i layer, le app e le istanze dello stack, e dettagli su ciascuna istanza come l'indirizzo IP.
- Attributi JSON personalizzati che contengono i dati forniti dall'utente e che possono avere lo stesso scopo dei contenitori di dati.

AWS OpsWorks Stacks installa una versione corrente degli attributi di configurazione e distribuzione dello stack su ogni istanza per ogni evento del ciclo di vita, prima di iniziare l'esecuzione Chef dell'evento. I dati sono disponibili per le ricette tramite la sintassi `node[:attribute][:child_attribute][...]` standard. Ad esempio, gli attributi di configurazione e distribuzione dello stack includono il nome dello stack `node[:opsworks][:stack][:name]`.

Il seguente segmento di uno dei libri di ricette predefiniti recupera il nome dello stack e lo utilizza per creare un file di configurazione.

```
template '/etc/ganglia/gmetad.conf' do
  source 'gmetad.conf.erb'
  mode '0644'
  variables :stack_name => node[:opsworks][:stack][:name]
  notifies :restart, "service[gmetad]"
end
```

Molti dei valori degli attributi di configurazione e distribuzione dello stack contengono più attributi. Devi eseguire iterazioni su questi attributi per recuperare le informazioni necessarie. L'esempio

seguente mostra un segmento di attributi di configurazione e distribuzione dello stack, rappresentati sotto forma di oggetto JSON per comodità. Contiene un attributo di livello superiore, `deploy`, che contiene un attributo per ogni app dello stack, denominato con il nome breve dell'app.

```
{
  ...
  "deploy": {
    "app1_shortname": {
      "document_root": "app1_root",
      "deploy_to": "deploy_directory",
      "application_type": "php",
      ...
    },
    "app2_shortname": {
      "document_root": "app2_root",
      ...
    }
  },
  ...
}
```

Ogni attributo dell'app contiene un set di attributi che caratterizzano l'app. Ad esempio, l'attributo `deploy_to` rappresenta la directory di distribuzione dell'app. Il seguente segmento imposta l'utente, il gruppo e il percorso della directory di distribuzione di ogni app.

```
node[:deploy].each do |application, deploy|
  opsworks_deploy_dir do
    user deploy[:user]
    group deploy[:group]
    path deploy[:deploy_to]
  end
  ...
end
```

Per ulteriori informazioni sugli attributi di configurazione e distribuzione dello stack, consulta Personalizzazione degli stack AWS OpsWorks. Per ulteriori informazioni sulle directory di distribuzione, consulta Ricette di ditribuzione.

Gli stack Chef 11.4 non supportano i contenitori di dati, ma puoi aggiungere dati arbitrari agli attributi di configurazione e distribuzione dello stack specificando dati JSON personalizzati. Le ricette potranno quindi accedere ai dati utilizzando la sintassi di nodo standard di Chef. Per ulteriori informazioni, consulta Utilizzo di un JSON personalizzato.

Se hai bisogno della funzionalità di un data bag crittografato, un'opzione è archiviare gli attributi sensibili in un luogo sicuro, ad esempio un bucket Amazon S3 privato. Le tue ricette possono quindi utilizzare l'SDK AWS Ruby, installato su tutte le istanze AWS OpsWorks Stacks, per scaricare i dati dal bucket.

> **Note**
>
> Ogni istanza Stacks ha un profilo di istanza. AWS OpsWorks Il ruolo IAM associato specifica a quali risorse AWS possono accedere le applicazioni in esecuzione sull'istanza. Affinché le tue ricette possano accedere a un bucket Amazon S3, la policy del ruolo deve includere un'istruzione simile alla seguente, che concede l'autorizzazione a recuperare i file da un bucket specificato.
>
> ```
> "Action": ["s3:GetObject"],
> "Effect": "Allow",
> "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
> ```
>
> Per ulteriori informazioni sui profili delle istanze, consulta Specificazione delle autorizzazioni per le app in esecuzione su istanze EC2 .

Migrazione di un stack Linux esistente in una nuova versione di Chef

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi utilizzare la console, l'API o la CLI di AWS OpsWorks Stacks per migrare i tuoi stack Linux a una versione più recente di Chef. Le ricette potrebbero tuttavia richiedere qualche modifica per poter essere compatibili con la versione più recente. Durante la preparazione della migrazione di uno stack, tieni presente le seguenti considerazioni.

- Non è possibile modificare le versioni AWS OpsWorks dello stack di Stacks da Chef 11 a Chef 12 modificando o clonando lo stack. Non è possibile eseguire l'aggiornamento della versione principale di Chef utilizzando la procedura descritta in questa sezione. Per ulteriori informazioni sulla transizione da Chef 11.10 a Chef 12, consulta Implementazione delle ricette: Chef 12.
- La transizione da una versione di Chef a un'altra richiede un certo numero di modifiche, alcune delle quali di maggiore entità.

  Per ulteriori informazioni sulla transizione da Chef 0.9 a Chef 11.4, consulta Migrazione in una nuova versione di Chef. Per ulteriori informazioni sulla transizione da Chef 11.4 a Chef 11.10, consulta Implementazione delle ricette: Chef 11.10. Per ulteriori informazioni sulla transizione da Chef 11.10 a Chef 12, consulta Implementazione delle ricette: Chef 12.
- Le esecuzioni di Chef utilizzano una versione di Ruby diversa sugli stack Chef 0.9 e Chef 11.4 (Ruby 1.8.7), sugli stack Chef 11.10 (Ruby 2.0.0) e sugli stack Chef 12 (Ruby 2.1.6).

  Per ulteriori informazioni, consulta Versioni di Ruby.
- Gli stack Chef 11.10 gestiscono l'installazione dei libri di ricette in modo diversi rispetto agli stack Chef 0.9 o Chef 11.4.

  Questa differenza potrebbe causare problemi durante la migrazione degli stack che utilizzano libri di ricette personalizzati in Chef 11.10. Per ulteriori informazioni, consulta Installazione e precedenza dei libri di ricette.

Le seguenti linee guida sono consigliate per la migrazione di uno stack Chef in una versione più recente di Chef:

Per eseguire la migrazione di uno stack in una versione più recente Chef

1. Clonare lo stack di produzione. Nella pagina Clone Stack (Clona stack), fare clic su Advanced>> (Avanzate>>) per visualizzare la sezione Configuration Management (Gestione configurazione e modificare Chef version (Versione Chef) immettendo la versione più aggiornata.

> **Note**
>
> Se si sta utilizzando uno stack Chef 0.9, non è possibile effettuare l'aggiornamento direttamente a Chef 11.10; è necessario effettuare prima l'aggiornamento a Chef 11.4. Se si desidera migrare lo stack a Chef 11.10 prima del test delle ricette, attendere 20 minuti per l'esecuzione dell'aggiornamento e quindi effettuare l'aggiornamento dello stack da 11.4 a 11.10.

2. Aggiungere le istanze ai livelli e testare le applicazioni e i libri di ricette dello stack clonano sul sistema di test o temporaneo. Per ulteriori informazioni, consulta la pagina relativa a Chef.
3. Quando i risultati del test sono soddisfacenti, procedere in uno dei seguenti modi:
   - Se questa è la versione desiderata di Chef, è possibile utilizzare lo stack clonato come stack di produzione oppure ripristinare la versione di Chef sullo stack di produzione.
   - Se si esegue la migrazione di uno stack Chef 0.9 a Chef 11.10 in due fasi, ripetere il processo per migrare lo stack da Chef 11.4 a Chef 11.10.

> **Note**
>
> Quando si testano le ricette, è possibile utilizzare SSH per connettersi all'istanza e quindi utilizzare il comando run_command della CLI di Instance Agent per eseguire le ricette associate ai vari eventi del ciclo di vita. La CLI dell'agente risulta particolarmente utile per il test delle ricette di impostazione (Configurazione) perché è possibile utilizzarla anche se la configurazione ha esito negativo e l'istanza non raggiunge lo stato online. È inoltre possibile utilizzare il comando Setup stack (Configura stack) per rieseguire le ricette di impostazione (Configurazione), ma tale comando è disponibile solo se la configurazione ha esito positivo e l'istanza è online.

È possibile aggiornare uno stack in esecuzione a una nuova versione di Chef.

Per aggiornare uno stack in esecuzione a una nuova versione di Chef

1. Modificare lo stack mediante la modifica dell'impostazione dell'opzione Chef version (Versione di Chef) per lo stack.

2. Salva le nuove impostazioni e attendi che AWS OpsWorks Stacks aggiorni le istanze, operazione che in genere richiede 15-20 minuti.

> **Important**
>
> AWS OpsWorks Stacks non sincronizza l'aggiornamento della versione di Chef con gli eventi del ciclo di vita. Se si desidera aggiornare la versione di Chef in uno stack di produzione, devi assicurarti che l'aggiornamento è stato completato prima del successivo evento del ciclo di vita. Se si verifica un evento, in genere un evento Deploy o Configure, l'agent dell'istanza aggiorna i libri di cucina personalizzati ed esegue le ricette assegnate all'evento, indipendentemente dal fatto che l'aggiornamento della versione sia completo o meno. Non esiste alcun modo diretto per determinare quando l'aggiornamento della versione è completato, se non controllare i log di distribuzione, dove viene inclusa la versione di Chef.

Versioni di Ruby

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ruby è installato su tutte le istanze di uno stack Linux. AWS OpsWorks Stacks installa un pacchetto Ruby su ogni istanza, che utilizza per eseguire le ricette di Chef e l'agent dell'istanza. AWS OpsWorks Stacks determina la versione di Ruby in base alla versione di Chef in esecuzione nello stack. Non tentare di modificare questa versione; questo può disabilitare l'agente di istanza.

AWS OpsWorks Stacks non installa un'applicazione Ruby eseguibile sugli stack di Windows. Il client Chef 12.2 viene fornito con Ruby 2.0.0 p451, ma l'eseguibile Ruby non viene aggiunto alla variabile di ambiente PATH delle istanze. Se vuoi usare questo eseguibile per l'esecuzione del codice Ruby, puoi trovarlo nel percorso `\opscode\chef\embedded\bin\ruby.exe` sul disco di Windows.

La tabella seguente riepiloga le versioni di AWS OpsWorks Stacks Ruby. Le versioni disponibili dell'applicazione Ruby variano inoltre a seconda del sistema operativo dell'istanza. Per ulteriori informazioni, incluse le versioni delle patch disponibili, consulta AWS OpsWorks Sistemi operativi impilati.

Versione di Chef	Versione di Ruby per Chef	Versioni applicazione Ruby disponibili
0.9 (c)	1.8.7	1.8.7(a), 1.9.3(e), 2.0.0
11.4 (c)	1.8.7	1.8.7(a), 1.9.3(e), 2.0.0, 2.1, 2.2.0, 2.3
11.10	2.0.0-p481	1.9.3(c, e), 2.0.0, 2.1, 2.2.0, 2.3, 2.6.1
12 (b)	2.1.6, 2.2.3	Nessuna
12.22 (d)	2.3.6	Nessuno

(a) Non disponibile con Amazon Linux versione 2014.09 e successive, Red Hat Enterprise Linux (RHEL) o Ubuntu 14.04 LTS.

(b) Disponibile solo su stack Linux.

(c) Non disponibile con RHEL.

(d) Disponibile solo su stack Windows. La versione principale è 12.2. La versione corrente minore è 12.22.

(e) La definizione come obsoleto è completata; il supporto è terminato.

La posizione di installazione dipende dalla versione di Chef:

- Le applicazioni utilizzano l'eseguibile `/usr/local/bin/ruby` per tutte le versioni di Chef.
- Per Chef 0.9 e 11.4, l'agente di istanza e le ricette Chef utilizzano l'eseguibile `/usr/bin/ruby`.
- Per Chef 11.10, l'agente di istanza e le ricette Chef utilizzano l'eseguibile `/opt/aws/opsworks/local/bin/ruby`.

Installazione di libri di ricette personalizzati

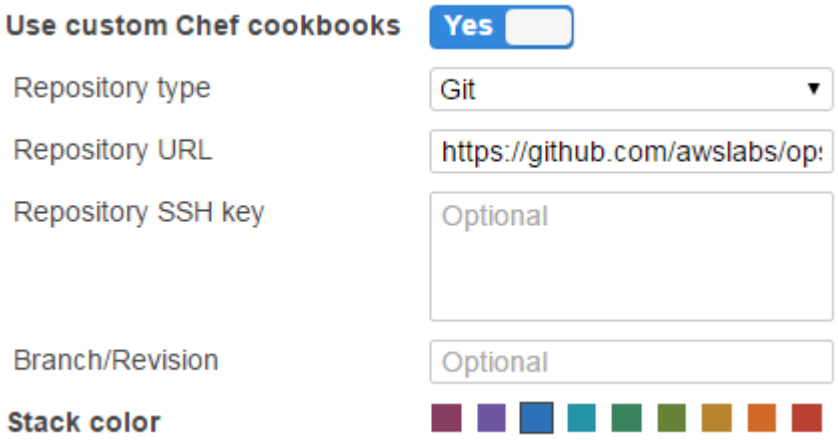
> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Affinché uno stack installi e utilizzi i libri di ricette personalizzati, devi configurare lo stack in modo che vengano abilitati i libri di ricette personalizzati, se non hai già eseguito questa configurazione. Devi quindi fornire l'URL dell'archivio e qualsiasi altra informazione correlata, ad esempio una password.

> **Important**
>
> Dopo aver configurato lo stack per supportare libri di cucina personalizzati, AWS OpsWorks Stacks installa automaticamente i tuoi libri di cucina su tutte le nuove istanze all'avvio. Tuttavia, devi indicare esplicitamente a AWS OpsWorks Stacks di installare libri di cucina nuovi o aggiornati su tutte le istanze esistenti eseguendo il comando Update Custom Cookbooks stack. Per ulteriori informazioni, consulta Aggiornamento dei libri di ricette personalizzati. Prima di abilitare la funzionalità Use custom Chef cookbooks (Utilizza i libri di ricette di Chef personalizzati) sullo stack, assicurarsi che i libri di ricette personalizzati e della community eseguiti supportino la versione di Chef utilizzata dallo stack.

Per configurare uno stack per i libri di ricette personalizzati

1. Nella pagina dello stack fare clic su Stack Settings (Impostazioni stack) per visualizzare la pagina Settings (Impostazioni). Fare clic su Edit (Modifica) per modificare le impostazioni.
2. Impostare Use custom Chef Cookbooks (Utilizza i libri di ricette di Chef personalizzati) su Yes (Sì).



3. Configurare i libri di ricette personalizzati.

Al termine, fare clic su Save (Salva) per salvare lo stack aggiornato.

Specifica di un archivio di libri di ricette personalizzati

Gli stack Linux possono installare libri di ricette personalizzati dai seguenti tipi di archivio:

- Archivi HTTP o Amazon S3.

  Possono essere pubblici o privati, ma Amazon S3 è in genere l'opzione preferita per un archivio privato.
- Gli archivi Git e Subversion forniscono il controllo del codice sorgente e la possibilità di avere più versioni.

Gli stack di Windows possono installare libri di cucina personalizzati dagli archivi Amazon S3 e dai repository Git.

Tutti i tipi di archivio dispongono dei seguenti campi obbligatori.

- Tipo di repository: il tipo di repository
- URL del repository: l'URL del repository

AWS OpsWorks Stacks supporta siti di repository Git ospitati pubblicamente come GitHubo Bitbucket, nonché server Git ospitati privatamente. Per gli archivi Git, devi utilizzare uno dei seguenti formati

di URL, a seconda che l'archivio sia pubblico o privato. Utilizza le stesse linee guida per gli URL dei moduli secondari Git.

Per un archivio Git pubblico, utilizza i protocolli HTTPS o Git di sola lettura:

- Git di sola lettura —. `git://github.com/amazonwebservices/opsworks-example-cookbooks.git`
- HTTPS —. `https://github.com/amazonwebservices/opsworks-example-cookbooks.git`

Per un archivio Git privato, devi utilizzare il formato in lettura/scrittura SSH, come illustrato negli esempi seguenti:

- Archivi Github —. `git@github.com:`*`project`*`/`*`repository`*
- Repository su un server Git — *`user`*`@`*`server`*`:`*`project`*`/`*`repository`*

Le restanti impostazioni variano a seconda del tipo di repository e sono descritte nelle seguenti sezioni.

Archivio HTTP

Selezionare Http Archive (Archivio Http) in Repository type (Tipo di repository) per visualizzare due impostazioni aggiuntive, che devono essere impostate se l'archivio è protetto mediante password.

- Nome utente: il tuo nome utente
- Password: la tua password

Archivio Amazon S3

Selezionare S3 Archive (Archivio S3) in Repository type (Tipo di repository) per visualizzare le seguenti impostazioni aggiuntive opzionali. AWS OpsWorks Gli stack possono accedere al tuo repository utilizzando EC2 i ruoli Amazon (autenticazione del gestore del sistema operativo host), indipendentemente dal fatto che utilizzi l'API AWS OpsWorks Stacks o la console.

- ID chiave di accesso: un ID della chiave di accesso AWS, ad esempioAKIAIOSFODNN7EXAMPLE.
- Chiave di accesso segreta: la chiave di accesso segreta AWS corrispondente, ad esempiowJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY.

Archivio Git

Se selezioni Git in Source Control (Controllo origine) vengono visualizzate le seguenti impostazioni aggiuntive opzionali:

Repository SSH key (Chiave SSH archivio)

Devi specificare una chiave SSH di distribuzione per accedere agli archivi Git privati. Per i moduli secondari Git, la chiave specificata deve avere accesso a tali moduli secondari. Per ulteriori informazioni, consulta Utilizzo di chiavi SSH di repository Git.

> **Important**
>
> La chiave SSH di distribuzione non può richiedere una password; AWS OpsWorks Stacks non ha modo di passarla.

Branch/Revision (Ramo/Versione)

Se il repository ha più rami, AWS OpsWorks Stacks scarica il ramo principale per impostazione predefinita. Per specificare un ramo particolare, inserisci il nome del ramo, l' SHA1 hash o il nome del tag. Per specificare un determinato commit, immetti l'ID di commit completo a 40 cifre esadecimali.

Archivio Subversion

Se selezioni Subversion in Source Control (Controllo origine) vengono visualizzate le seguenti impostazioni aggiuntive:

- Nome utente: il tuo nome utente, per gli archivi privati.
- Password: la tua password, per gli archivi privati.
- Revisione — [Facoltativo] Il nome della revisione, se sono presenti più revisioni.

Per specificare un ramo o un tag, devi modificare l'URL del repository, ad esempio: **`http://repository_domain/repos/myapp/branches/my-apps-branch`** o **`http://repository_domain_name/repos/calc/myapp/my-apps-tag`**.

Aggiornamento dei libri di ricette personalizzati

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Quando fornisci a AWS OpsWorks Stacks libri di cucina personalizzati, le ricette di installazione integrate creano una cache locale su ogni istanza appena avviata e scaricano i libri di cucina nella cache. AWS OpsWorks Stacks esegue quindi le ricette dalla cache, non dal repository. Se modifichi i libri di cucina personalizzati nel repository, devi assicurarti che i libri di cucina aggiornati siano installati nelle cache locali delle tue istanze. AWS OpsWorks Stacks distribuisce automaticamente i libri di cucina più recenti su nuove istanze quando vengono avviati. Per le istanze esistenti, tuttavia, la situazione è diversa:

- Devi distribuire manualmente i libri di ricette personalizzati aggiornati per le istanze online.
- Non devi distribuire i libri di ricette personalizzati aggiornati per le istanze basate sugli instance store offline, tra cui istanze basate sul carico e sul tempo.

  AWS OpsWorks Stacks distribuisce automaticamente i libri di cucina correnti al riavvio delle istanze.
- Devi avviare le istanze offline 24 ore su 24, 7 giorni su 7 supportate da EBS, le quali non sono basate sul carico o sul tempo.
- Non puoi avviare le istanze offline supportate da EBS basate sul carico e sul tempo, perciò l'approccio più semplice è quello di eliminare le istanze offline e aggiungere nuove istanze per sostituirle.

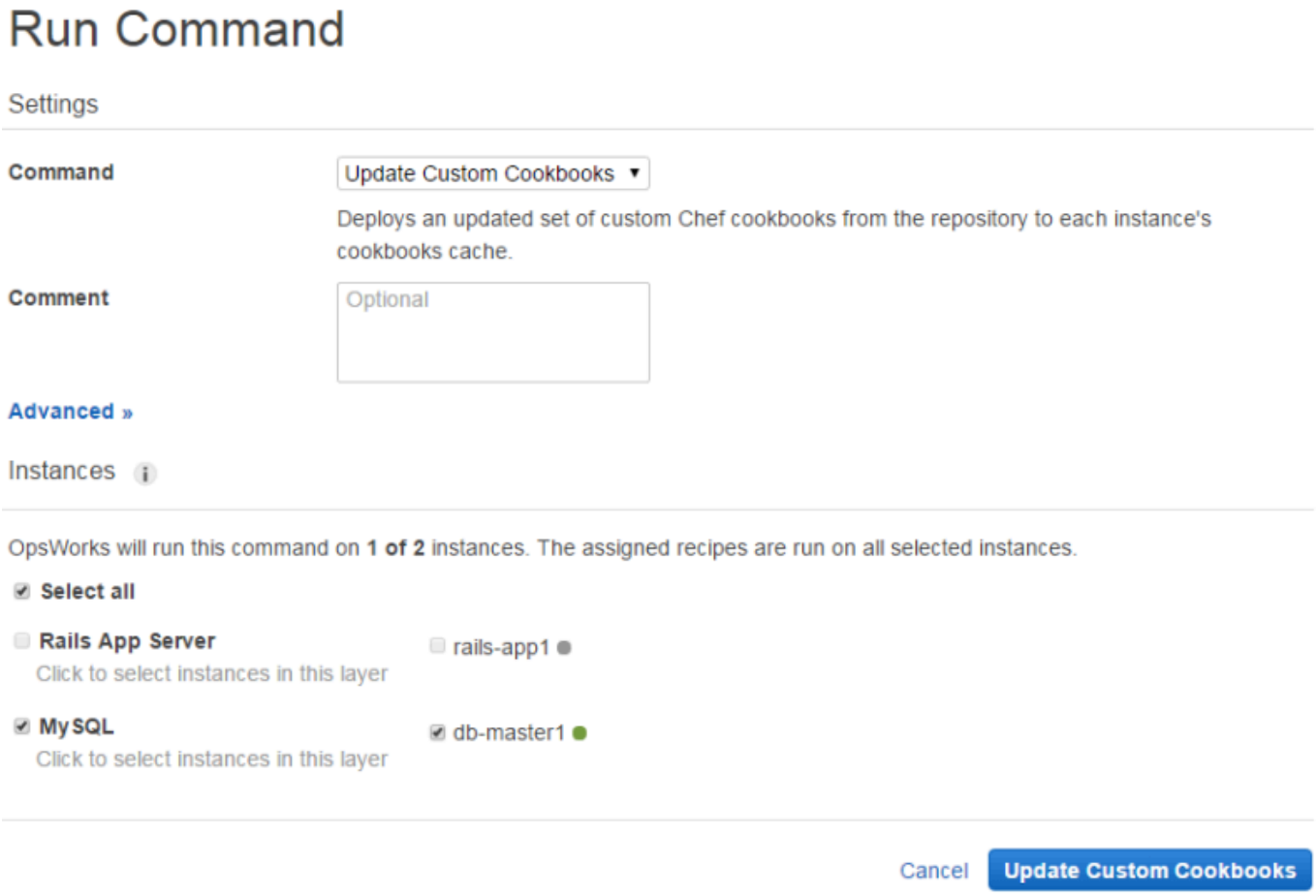
  Poiché ora sono nuove istanze, AWS OpsWorks Stacks distribuisce automaticamente gli attuali libri di cucina personalizzati all'avvio delle istanze.

Per aggiornare manualmente i libri di ricette personalizzati

1. Aggiorna il tuo repository con i libri di cucina modificati. AWS OpsWorks Stacks utilizza l'URL della cache che hai fornito quando hai originariamente installato i libri di cucina, quindi il nome del file principale del libro di cucina, la posizione del repository e i diritti di accesso non devono cambiare.

   - Per i repository Amazon S3 o HTTP, sostituisci il file.zip originale con un nuovo file.zip con lo stesso nome.
   - For gli archivi Git o Subversion, modifica le impostazioni del tuo stack per modificare il campo Branch/Revision (Ramo/Versione) per la nuova versione.

2. Nella pagina dello stack, fai clic su Run Command (Esegui comando) e seleziona il comando Update Custom Cookbooks (Aggiorna libri di ricette personalizzati).



3. Se lo desideri, aggiungi un commento.
4. Facoltativamente, specifica un oggetto JSON personalizzato per il comando per aggiungere attributi personalizzati alla configurazione dello stack e agli attributi di distribuzione che Stacks

installa sulle istanze. AWS OpsWorks Per ulteriori informazioni, consultare Utilizzo di un JSON personalizzato e Sostituzione degli attributi.

5. Per impostazione predefinita, AWS OpsWorks Stacks aggiorna i libri di cucina su ogni istanza. Per specificare quali istanze aggiornare, seleziona le istanze appropriate dall'elenco al termine della pagina. Per selezionare ogni istanza in un livello, seleziona la casella di controllo del livello appropriato dalla colonna a sinistra.
6. Fai clic su Aggiorna libri di cucina personalizzati per installare i ricettari aggiornati. AWS OpsWorks Stacks elimina i libri di cucina personalizzati memorizzati nella cache nelle istanze specificate e installa i nuovi libri di cucina dal repository.

> **Note**
>
> Questa procedura è necessaria solo per le istanze esistenti, che dispongono di vecchie versioni dei libri di ricette nelle loro cache. Se successivamente aggiungi istanze a un layer, AWS OpsWorks Stacks distribuisce i libri di cucina attualmente presenti nel repository in modo che ottengano automaticamente la versione più recente.

Esecuzione di ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

È possibile eseguire le ricette in due modi:

- Automaticamente, assegnando le ricette all'evento del ciclo di vita del livello appropriato.
- Manualmente, eseguendo il comando dello stack Execute Recipes (Esegui ricette) oppure utilizzando l'interfaccia a riga di comando dell'agente.

Argomenti

- AWS OpsWorks Eventi del ciclo di vita degli stack
- Esecuzione automatica di ricette
- Esecuzione manuale delle ricette

AWS OpsWorks Eventi del ciclo di vita degli stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ogni livello ha un set di cinque eventi del ciclo di vita, ognuno dei quali ha un set di ricette associato specifico per il livello. Quando si verifica un evento in un'istanza di un livello, AWS OpsWorks Stacks esegue automaticamente il set di ricette appropriato. Per fornire una risposta personalizzata a questi eventi, implementa ricette personalizzate e assegnale agli eventi appropriati per ogni livello. AWS OpsWorks Stacks esegue queste ricette dopo le ricette integrate dell'evento.

Setup

Questo evento si verifica dopo che un'istanza avviata ha completato la procedura di avvio. È inoltre possibile attivare manualmente l'Setupevento utilizzando il comando Setup stack. AWS OpsWorks Stacks esegue ricette che configurano l'istanza in base al relativo livello. Ad esempio, se l'istanza è un membro del livello Rails App Server, le Setup ricette installano Apache, Ruby Enterprise Edition, Passenger e Ruby on Rails.

> **Note**
>
> Un evento Setup (Impostazione) imposta un'istanza come fuori servizio. Poiché un'istanza non si trova nello stato Online quando viene eseguito l'evento del ciclo di vita Setup (Impostazione), le istanze su cui esegui gli eventi Setup (Impostazione) vengono rimosse da un sistema di bilanciamento del carico.

Configure

Questo evento si verifica su tutte le istanze dello stack in presenza di una delle condizioni seguenti:

- Un'istanza passa allo stato online o esce da tale stato.
- Un indirizzo IP elastico viene associato a un'istanza o ne viene eliminata l'associazione da un'istanza.
- Puoi collegare un sistema di bilanciamento del carico di Elastic Load Balancing a un livello o scollegarlo da un livello.

Ad esempio, supponiamo che lo stack contenga le istanze A, B e C e che tu inizi una nuova istanza, D. Dopo che D ha terminato l'esecuzione delle ricette di configurazione, AWS OpsWorks Stacks attiva l'Configureevento su A, B, C e D. Se successivamente interrompi A, AWS OpsWorks Stacks attiva l'Configureevento su B, C e D. AWS OpsWorks Stacks risponde all'Configureevento eseguendo le Configure ricette di ogni livello, che aggiornano la configurazione delle istanze per riflettere la corrente set di istanze online. L'evento Configure rappresenta quindi una buona occasione per rigenerare i file di configurazione. Ad esempio, le HAProxy Configure ricette riconfigurano il load balancer per adattarsi a eventuali modifiche nel set di istanze dell'application server online.

Puoi anche avviare manualmente l'evento Configure (Configurazione) utilizzando il comando dello stack Configure (Configurazione).

Deploy

Questo evento si verifica quando esegui un comando Deploy (Distribuzione), in genere per distribuire un'applicazione in un set di istanze di server applicazioni. Le istanze eseguono le ricette che distribuiscono l'applicazione e i file correlati dal repository alle istanze del livello. Ad esempio, per le istanze Rails Application Server, le ricette Deploy consultano una determinata applicazione Ruby e comunicano a Phusion Passenger di ricaricarla. Puoi inoltre eseguire Deploy in altre istanze per permettere, ad esempio, di aggiornare la configurazione in base all'app appena distribuita.

> **Note**
>
> L'evento Setup (Impostazione) include Deploy (Distribuzione) ed esegue le ricette di Deploy (Distribuzione) al termine dell'impostazione.

Undeploy

Questo evento si verifica quando elimini un'app o esegui un comando Undeploy per rimuovere un'app da un set di istanze server dell'applicazione. Le istanze specificate eseguono le ricette per rimuovere tutte le versioni delle applicazioni ed eseguire tutte le operazioni di pulizia richieste.

Shutdown

Questo evento si verifica dopo aver ordinato a AWS OpsWorks Stacks di chiudere un'istanza ma prima che l' EC2 istanza Amazon associata venga effettivamente terminata. AWS OpsWorks Stacks esegue ricette per eseguire attività di pulizia come la chiusura dei servizi.

Se hai collegato un load balancer Elastic Load Balancing al layer e hai abilitato il supporto per il drenaggio delle connessioni, AWS OpsWorks Stacks attende il completamento del drenaggio della connessione prima di attivare l'evento. Shutdown

Dopo aver attivato un Shutdown evento, AWS OpsWorks Stacks concede alle Shutdown ricette un determinato periodo di tempo per eseguire le proprie attività, quindi interrompe o chiude l'istanza Amazon. EC2 Il valore predefinito di timeout per Shutdown è di 120 secondi. Se le ricette Shutdown richiedono più tempo, puoi modificare la configurazione del livello per cambiare il valore di timeout. Per ulteriori informazioni sull'evento di un'istanza di tipo Shutdown, consulta Arresto di un'istanza.

> **Note**
>
> Riavviando un'istanza non si attivano eventi del ciclo di vita.

Per ulteriori informazioni sui comandi delle app Deploy e Undeploy consulta Distribuzione di app.

Dopo che un'istanza avviata ha completato la procedura di avvio, la sequenza di avvio rimanente è la seguente:

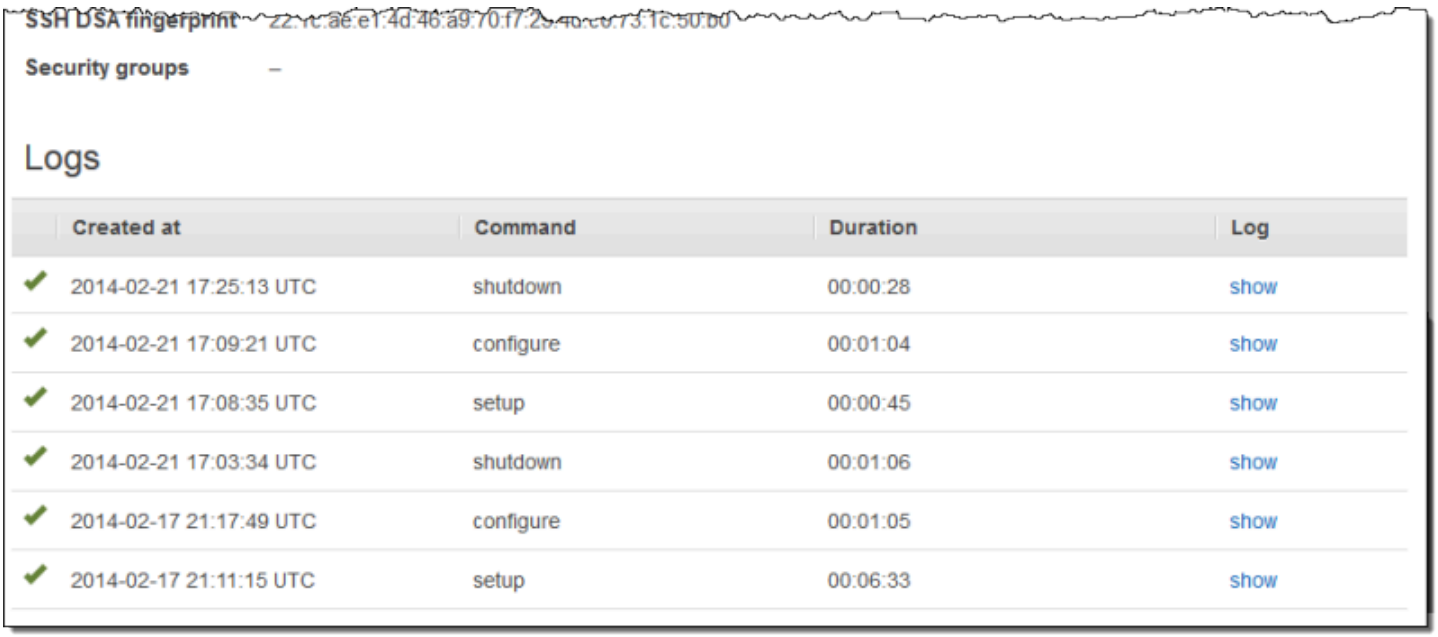
1. AWS OpsWorks Stacks esegue le Setup ricette integrate dell'istanza, seguite da eventuali ricette personalizzate. Setup
2. AWS OpsWorks Stacks esegue le Deploy ricette integrate dell'istanza, seguite da eventuali ricette personalizzateDeploy.

   L'istanza è ora online.

3. AWS OpsWorks Stacks attiva un Configure evento su tutte le istanze dello stack, inclusa l'istanza appena avviata.

   AWS OpsWorks Stacks esegue le Configure ricette integrate delle istanze, seguite da eventuali ricette personalizzate. Configure

> **Note**
>
> Per visualizzare gli eventi del ciclo di vita che si sono verificati in una determinata istanza, passa alla pagina Instances (Istanze) e fai clic sul nome dell'istanza per aprire la relativa pagina dei dettagli. L'elenco di eventi si trova nella sezione Logs (Log) nella parte inferiore della pagina. Puoi fare clic su show (mostra) nella colonna Log per esaminare i log di Chef per un evento. Vengono fornite informazioni dettagliate su come è stato gestito l'evento, tra cui l'indicazione delle ricette eseguite. Per ulteriori informazioni su come interpretare i log di Chef, consulta Log di Chef.



Per ogni evento del ciclo di vita, AWS OpsWorks Stacks installa un set di attributi di configurazione e distribuzione dello stack su ogni istanza che contiene lo stato corrente dello stack e, in caso di eventi, informazioni sulla distribuzione. Deploy Gli attributi includono le informazioni sulle istanze disponibili, i loro indirizzi IP e così via. Per ulteriori informazioni, consulta Attributi di configurazione e distribuzione dello stack.

> **Note**
>
> L'avvio o l'arresto di un numero elevato di istanze nello stesso momento può generare la creazione di un numero elevato di eventi Configure. Per evitare elaborazioni non necessarie, AWS OpsWorks Stacks risponde solo all'ultimo evento. Gli attributi di configurazione e distribuzione dello stack di tale evento contengono tutte le informazioni necessarie per aggiornare le istanze dello stack per l'intero set di modifiche. Ciò elimina la necessità di elaborare anche Configure gli eventi precedenti. AWS OpsWorks Stacks etichetta gli Configure eventi non elaborati come sostituiti.

Esecuzione automatica di ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ogni livello ha un set di ricette predefinite assegnate a ogni evento del ciclo di vita, anche se alcuni livelli non includono le ricette di annullamento della distribuzione. Quando si verifica un evento del ciclo di vita su un'istanza, AWS OpsWorks Stacks esegue il set di ricette appropriato per il layer associato.

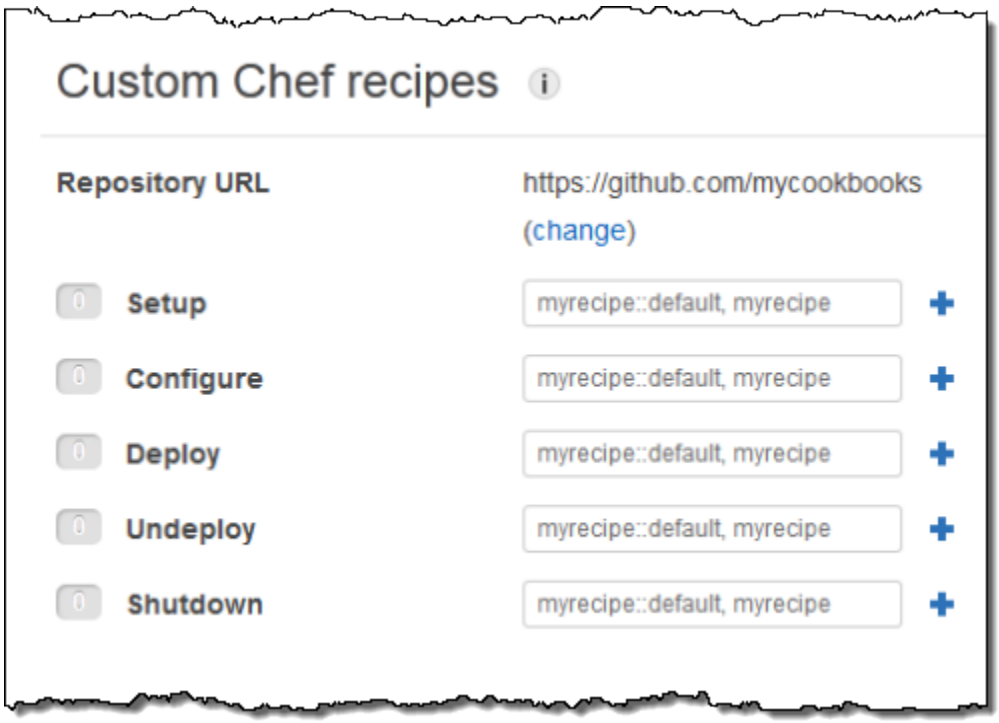
Se hai installato libri di cucina personalizzati, puoi fare in modo che AWS OpsWorks Stacks esegua automaticamente alcune o tutte le ricette assegnando ciascuna ricetta a un evento del ciclo di vita di un layer. Dopo che si è verificato un evento, AWS OpsWorks Stacks esegue le ricette personalizzate specificate dopo le ricette integrate del layer.

Per assegnare ricette personalizzate agli eventi del livello

1. Nella pagina Layers (Livelli), per il livello appropriato, fare clic su Recipes (Ricette), quindi fare clic su Edit (Modifica). Se non sono ancora stati abilitati libri di ricette personalizzati, fare clic su configure cookbooks (configura libri di ricette) per aprire la pagina Settings (Impostazioni) dello stack. Impostare Use custom Chef Cookbooks (Utilizza libri di ricette di Chef personalizzati) su

Yes (Sì) e fornire le informazioni sul repository del libro di ricette. Fare quindi clic su Save (Salva) e tornare alla pagina di modifica per la scheda Recipes (Ricette). Per ulteriori informazioni, consulta Installazione di libri di ricette personalizzati.

2. Nella scheda Recipes (Ricette) immettere ogni ricetta personalizzata nel campo dell'evento appropriato e fare clic su + per aggiungerla all'elenco. Specificate una ricetta come segue:*cookbook*: *somerecipe* (omettete l'`.rb`estensione).



Quando avvii una nuova istanza, AWS OpsWorks Stacks esegue automaticamente le ricette personalizzate per ogni evento, dopo aver eseguito le ricette standard.

> **Note**
>
> Le ricette personalizzate vengono eseguite nell'ordine con cui vengono immesse nella console. In alternativa, per controllare l'ordine di esecuzione, è possibile implementare una meta-ricetta che esegue le ricette nell'ordine corretto.

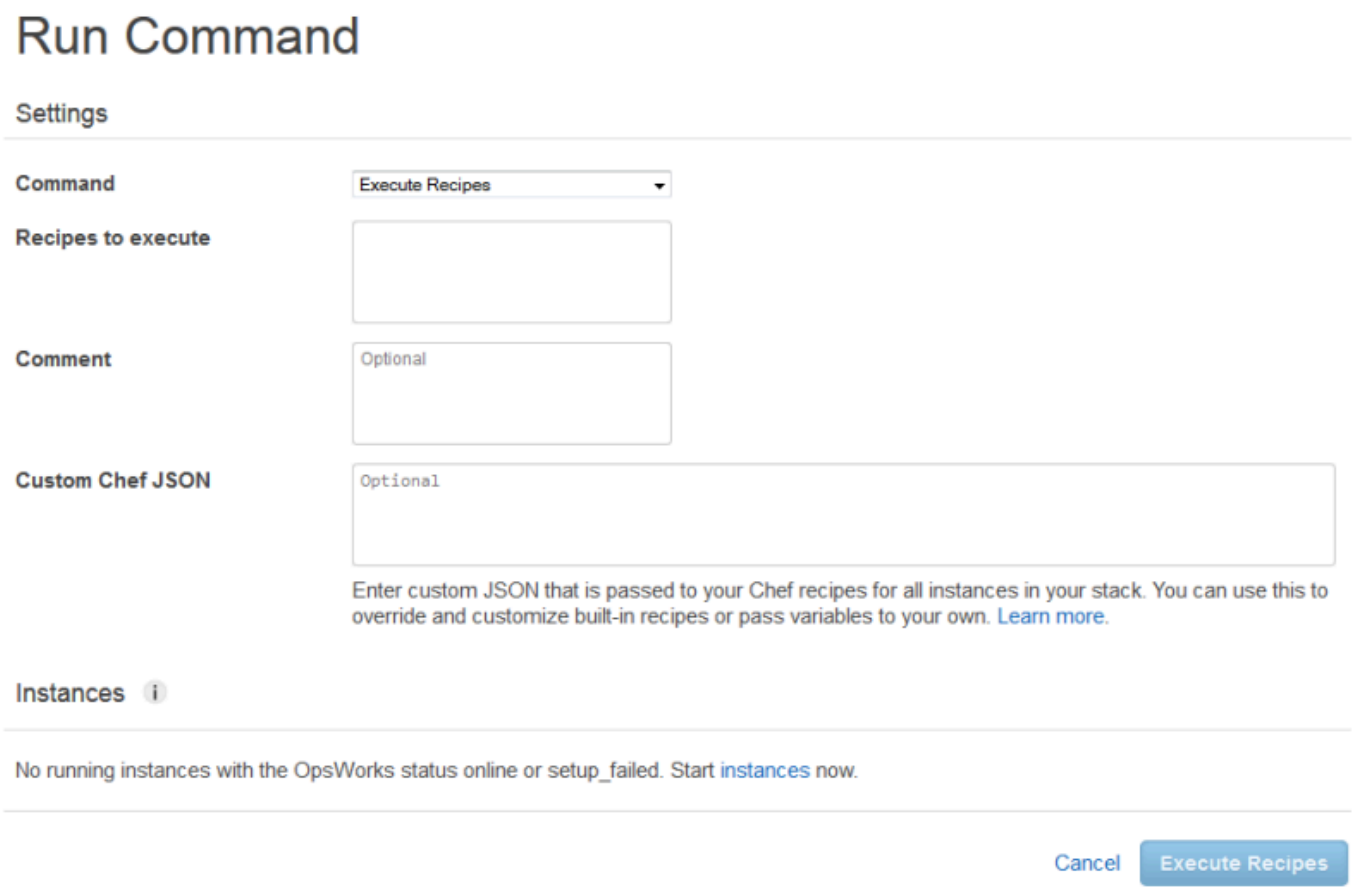
Esecuzione manuale delle ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Anche se le ricette in genere vengono eseguite automaticamente in risposta agli eventi del ciclo di vita, è possibile eseguirle manualmente in qualsiasi momento in tutte le istanze dello stack o in alcune di esse. Questa caratteristica viene in genere utilizzata per le attività non idonee per essere mappate a un evento del ciclo di vita, ad esempio il backup delle istanze. Per eseguire una ricetta personalizzata manualmente, è necessario che la ricetta si trovi in uno dei libri di ricette personalizzati, ma non è necessario che sia assegnata a un evento del ciclo di vita. Quando esegui una ricetta manualmente, AWS OpsWorks Stacks installa gli stessi `deploy` attributi che utilizza per un evento Deploy.

Per eseguire manualmente le ricette nelle istanze dello stack

1. Nella pagina Stack fare clic su Run command (Esegui comando). Per Command (Comando) selezionare Execute Recipes (Esegui ricette).



2. Immettete le ricette da eseguire nella casella Ricette da eseguire utilizzando il formato standard*cookbookname*::. *recipename* Utilizzare le virgole per separare più ricette. Le ricette verranno eseguite nell'ordine in cui sono elencate.
3. Facoltativamente, utilizzare la casella Custom Chef JSON (JSON Chef personalizzato) per aggiungere un oggetto JSON personalizzato che definisce gli attributi personalizzati che verranno uniti agli attributi di configurazione e distribuzione dello stack installati nelle istanze. Per ulteriori informazioni sull'utilizzo di oggetti JSON personalizzati, consulta Utilizzo di un JSON personalizzato e Sostituzione degli attributi.
4. In Istanze, seleziona le istanze su cui AWS OpsWorks Stacks deve eseguire le ricette.

Quando si verifica un evento del ciclo di vita, l'agente AWS OpsWorks Stacks riceve un comando per eseguire le ricette associate. È possibile eseguire manualmente questi comandi in una determinata istanza utilizzando il comando dello stack appropriato oppure utilizzando il comando run_command dell'interfaccia a riga di comando dell'agente. Per ulteriori informazioni su come utilizzare l'interfaccia a riga di comando dell'agente, consulta AWS OpsWorks CLI di Stacks Agent.

Gestione delle risorse

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

La pagina Risorse ti consente di utilizzare l'indirizzo IP elastico del tuo account, il volume Amazon EBS o le risorse dell'istanza Amazon RDS in uno AWS OpsWorks stack Stacks. Puoi utilizzare Resources (Risorse) per eseguire le operazioni seguenti:

- Registrare una risorsa con uno stack, che ti consente di collegare la risorsa a una delle istanze dello stack.
- Collegare una risorsa a una delle istanze dello stack.
- Spostare una risorsa da un'istanza a un'altra.
- Scollegare una risorsa da un'istanza. La risorsa rimane registrata e può essere collegata a un'altra istanza.
- Annullare la registrazione di una risorsa. Una risorsa non registrata non può essere utilizzata da AWS OpsWorks Stacks, ma rimane nel tuo account a meno che non la elimini e può essere registrata in un altro stack.

Vengono applicati i vincoli seguenti:

- Non è possibile collegare volumi Amazon EBS registrati a istanze Windows.
- La pagina Risorse gestisce i volumi Amazon EBS standard, PIOPS, Throughput Optimized, Cold HDD o General Purpose (SSD), ma non gli array RAID.
- I volumi Amazon EBS devono essere in formato xfs.

  AWS OpsWorks Stacks non supporta altri formati di file, come ext4. Per ulteriori informazioni sulla preparazione dei volumi Amazon EBS, consulta Making an Amazon EBS Volume Available for Use.
- Non puoi collegare un volume Amazon EBS a o scollegarlo da un'istanza in esecuzione.

Puoi intervenire solo sulle istanze offline. Ad esempio, puoi registrare un volume in uso con uno stack e collegarlo a un'istanza offline, ma devi arrestare l'istanza originale e scollegare il volume prima di avviare la nuova istanza. In alternativa, il processo di avvio avrà esito negativo.

- Tutte le risorse registrate sono gestite esclusivamente in. AWS OpsWorks Ciò può sostituire le proprietà del ciclo di vita delle risorse, ad esempio per i `DeleteOnTermination` volumi. EC2
- Puoi collegare un indirizzo IP elastico a un'istanza in esecuzione o scollegarlo.

  Puoi intervenire su istanze online o offline. Ad esempio, puoi registrare un indirizzo in uso e assegnarlo a un'istanza in esecuzione, e AWS OpsWorks Stacks riassegnerà automaticamente l'indirizzo.
- Per registrare risorse o annullarne la registrazione, la policy di IAM deve concedere le autorizzazioni per le seguenti azioni:

Volumi Amazon EBS	Indirizzi IP elastici	Istanze Amazon RDS
RegisterVolume	RegisterElasticIp	RegisterRdsDbInstance
UpdateVolume	UpdateElasticIp	UpdateRdsDbInstance
DeregisterVolume	DeregisterElasticIp	DeregisterRdsDbInstance

Il livello Manage permissions (Gestire autorizzazioni) concede autorizzazioni per tutte queste azioni. Per evitare che un utente Manage registri determinate risorse o ne annulli la registrazione, modifica la policy IAM per rifiutare le autorizzazioni per le azioni appropriate. Per ulteriori informazioni, consulta Sicurezza e autorizzazioni.

Argomenti

- Registrazione di risorse con uno stack
- Collegamento e spostamento di risorse
- Scollegamento delle risorse
- Annullamento della registrazione delle risorse

Registrazione di risorse con uno stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

I volumi Amazon EBS o gli indirizzi IP elastici devono essere registrati in uno stack prima di poterli collegare alle istanze. Quando AWS OpsWorks Stacks crea risorse per uno stack, queste vengono registrate automaticamente in quello stack. Per utilizzare le risorse create esternamente, devi registrarle in modo esplicito. Tieni presente quanto segue:

- Puoi registrare una risorsa con un solo stack alla volta.
- Quando elimini uno stack, AWS OpsWorks Stacks annulla la registrazione di tutte le risorse.

Argomenti

- Registrazione di volumi Amazon EBS con uno stack
- Registrazione di indirizzi IP elastici con uno stack
- Registrazione di istanze Amazon RDS con uno stack

Registrazione di volumi Amazon EBS con uno stack

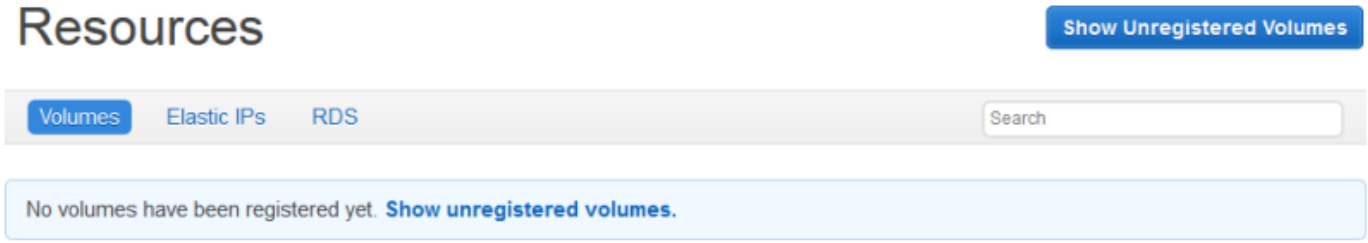
> **Note**
>
> Questa risorsa può essere utilizzata solo con gli stack Linux. Sebbene sia possibile registrare un volume Amazon EBS con uno stack Windows, non è possibile collegarlo a un'istanza.

Puoi utilizzare la pagina Risorse per registrare un volume Amazon EBS con uno stack, nel rispetto dei seguenti vincoli:

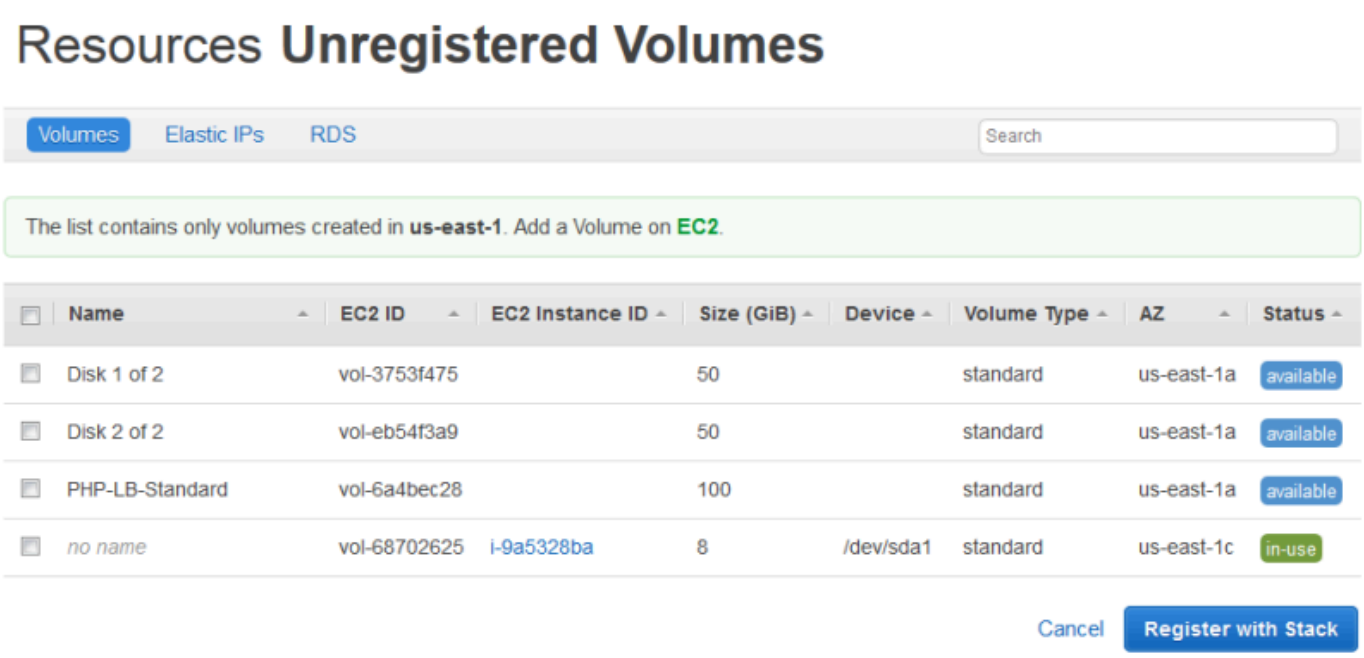
- I volumi Amazon EBS collegati e non root devono essere HDD standard, Throughput Optimized HDD, Cold HDD, PIOPS o General Purpose (SSD), ma non un array RAID. Per ulteriori informazioni sulle dimensioni massima e minima dei volumi, consulta Volumi EBS in questa guida.
- I volumi devono essere nel formato XFS.
- AWS OpsWorks Stacks non supporta altri formati di file, come il quarto file system esteso (ext4), per volumi Amazon EBS non root. Per ulteriori informazioni sulla preparazione dei volumi Amazon EBS, consulta Making an Amazon EBS Volume Available for Use. L'esempio in questo argomento descrive come creare un volume basato su ext4, tuttavia puoi utilizzare la stessa procedura per i volumi basati su XFS.

Per registrare un volume Amazon EBS

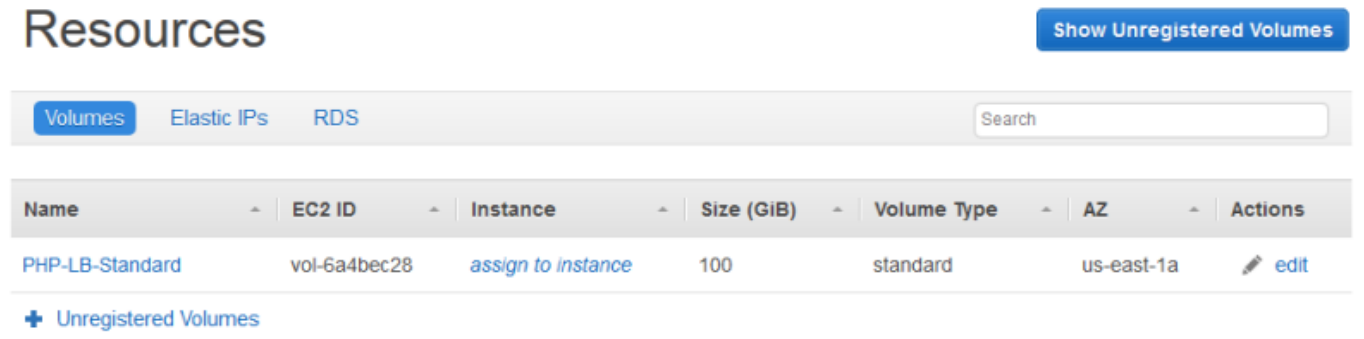
1. Aprire lo stack desiderato e fare clic su Resources (Risorse) nel riquadro di navigazione.
2. Fai clic su Volumi per visualizzare i volumi Amazon EBS disponibili. Inizialmente, lo stack non ha volumi registrati, come mostrato nell'illustrazione seguente.



3. Fai clic su Mostra volumi non registrati per visualizzare i volumi Amazon EBS del tuo account che si trovano nella regione dello stack e, se applicabile, il VPC dello stack. La colonna Status (Stato) indica se i volumi sono disponibili per l'uso. Volume Type (Tipo di volume) indica se il volume è standard (`standard`), General Purpose SSD (`gp2`), PIOPS (`io1`, seguito da valore IOPS per disco tra parentesi), Throughput Optimized HDD (`st1`) o Cold HDD (`sc1`).



4. Selezionare i volumi appropriati e fare clic su Register to Stack (Registra a stack). Nella pagina Resources (Risorse) sono ora elencati i nuovi volumi registrati.



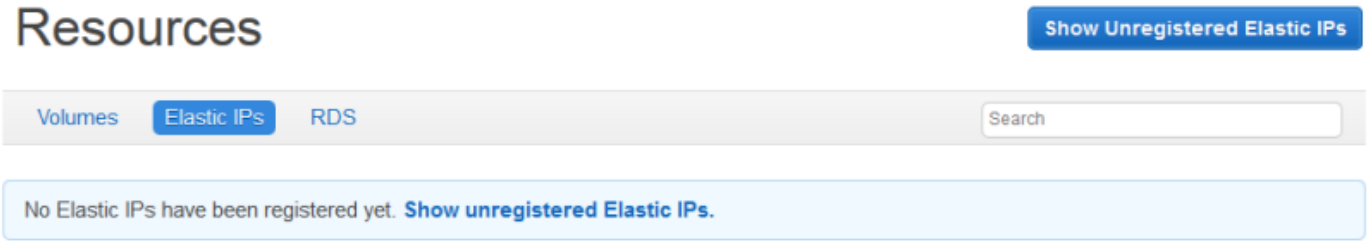
   Per registrare volumi aggiuntivi, fare clic su Show Unregistered Volumes (Mostra volumi non registrati) o su + Unregistered Volumes (+ Volumi non registrati) e ripetere questa procedura.

Registrazione di indirizzi IP elastici con uno stack

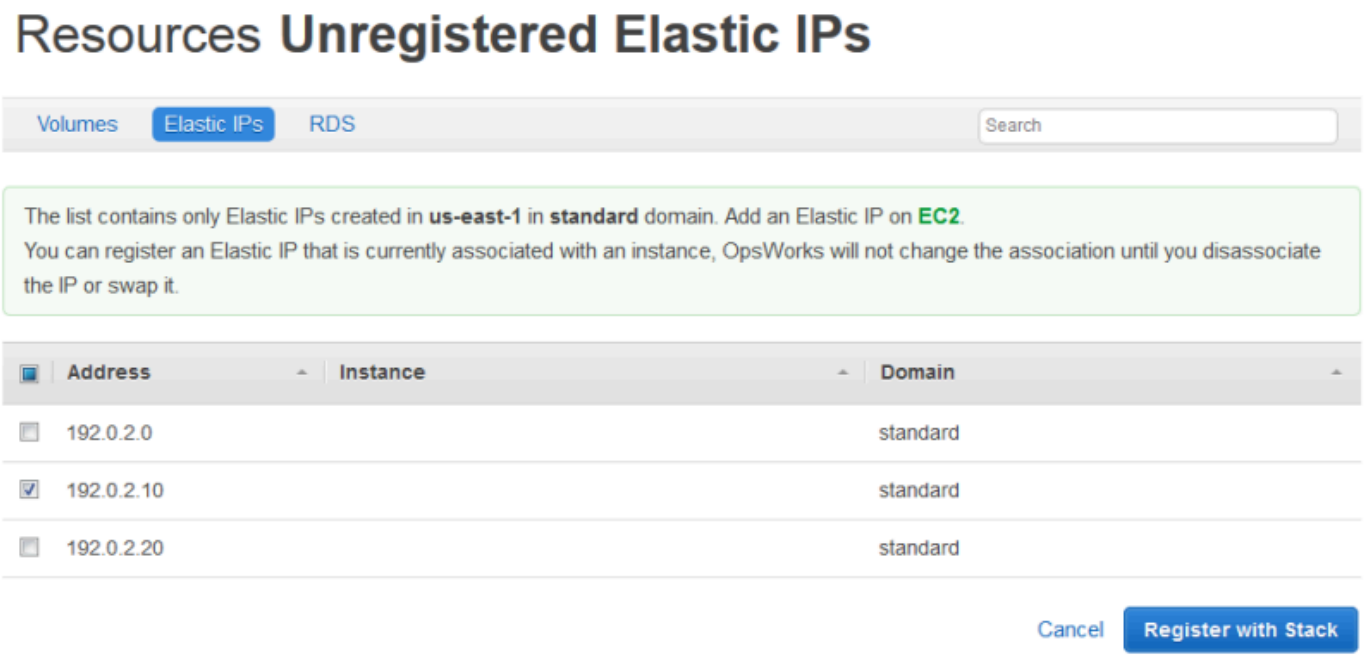
Utilizza la procedura seguente per registrare gli indirizzi IP elastici.

Per registrare un indirizzo IP elastico

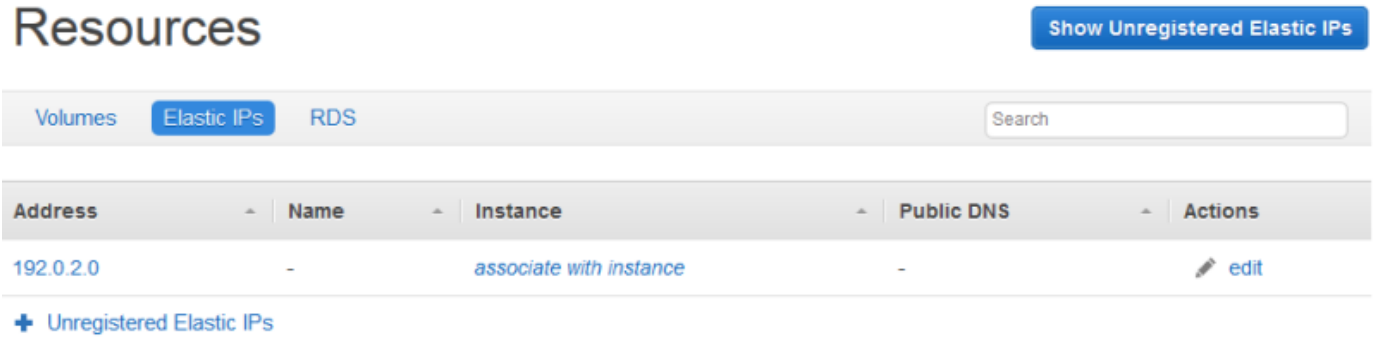
1. Apri la pagina Risorse dello stack e fai clic su Elastic IPs per visualizzare gli indirizzi IP elastici disponibili. Inizialmente, lo stack non ha indirizzi registrati, come mostrato nell'illustrazione seguente.



2. Fai clic su Mostra Elastic non registrato IPs per visualizzare gli indirizzi IP Elastic disponibili nel tuo account che si trovano nella regione dello stack.



3. Selezionare gli indirizzi appropriati e fare clic su Register to Stack (Registra a stack). Viene visualizzata di nuovo la pagina Resources (Risorse), che ora riporta i nuovi indirizzi registrati.



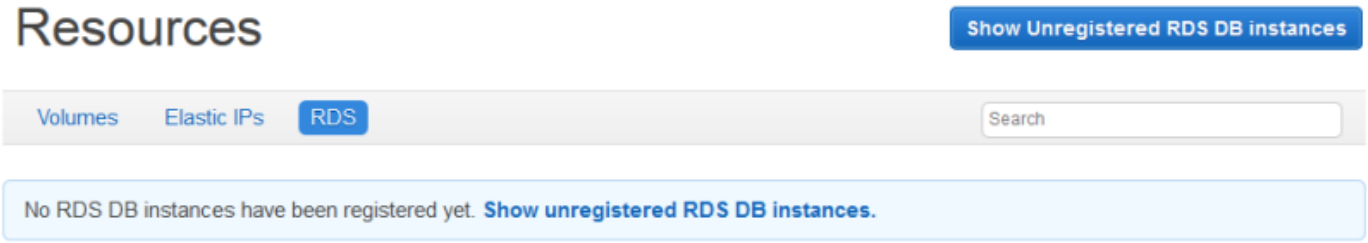
Per registrare indirizzi aggiuntivi, fai clic su Mostra elastico non registrato IPs o su + Elastico non registrato IPs e ripeti questa procedura.

Registrazione di istanze Amazon RDS con uno stack

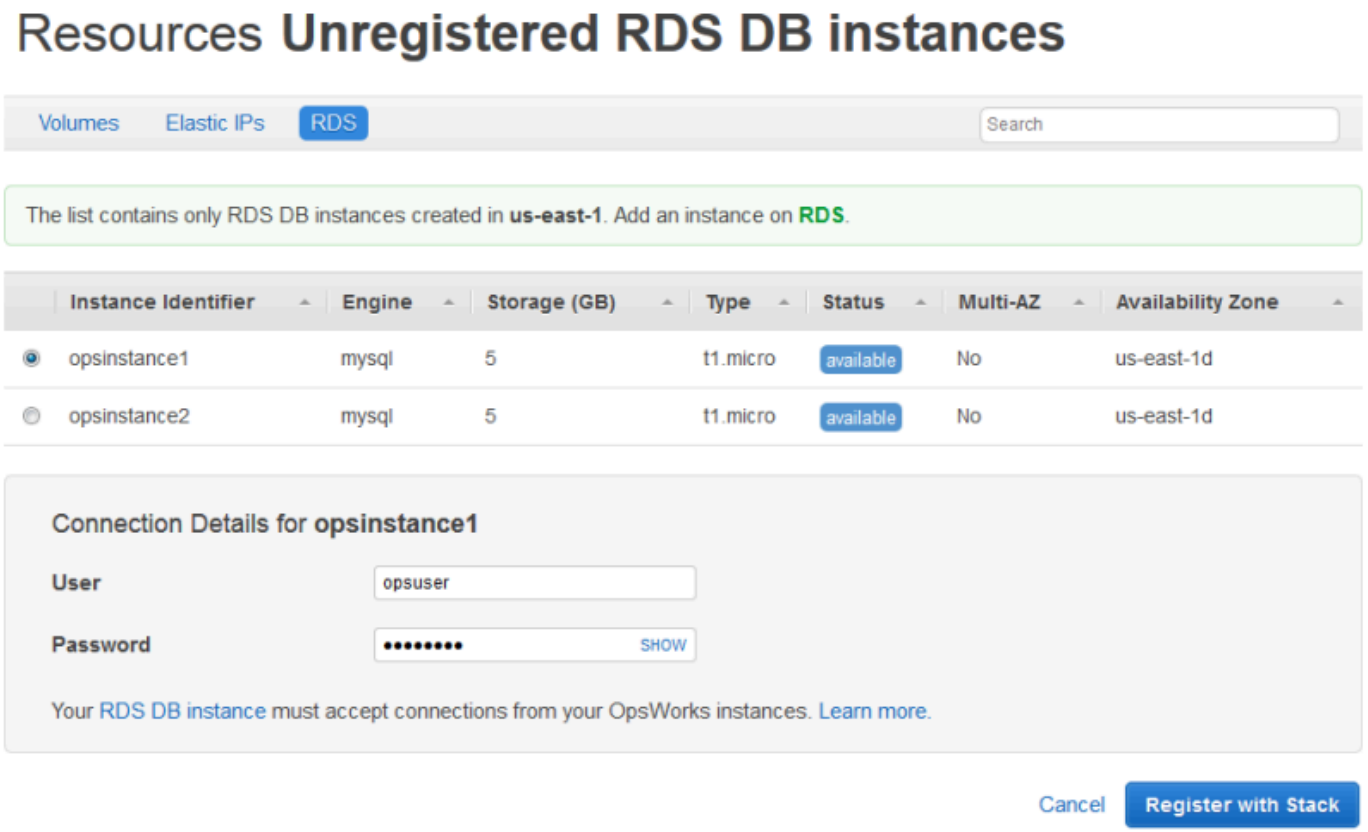
Utilizza la seguente procedura per registrare le istanze Amazon RDS.

Per registrare un'istanza Amazon RDS

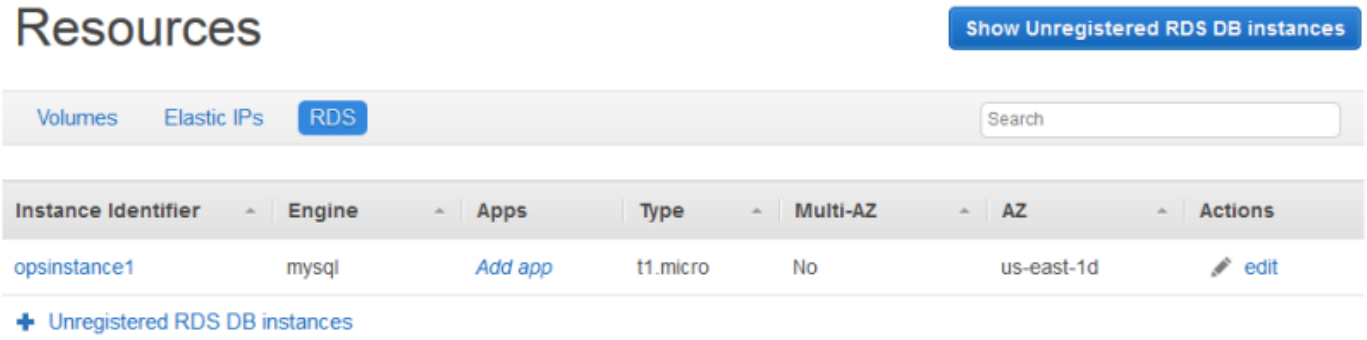
1. Apri la pagina Risorse dello stack e fai clic su RDS per visualizzare le istanze Amazon RDS disponibili. Inizialmente, lo stack non ha istanze registrate, come mostrato nell'illustrazione seguente.



2. Fai clic su Mostra istanze DB RDS non registrate per visualizzare le istanze Amazon RDS disponibili nel tuo account che si trovano nella regione dello stack.



3. Selezionare l'istanza appropriata, inserire i valori di utente master e password per User (Utente) e Password e fare clic su Register to Stack (Registra a stack). Viene visualizzata di nuovo la pagina Resources (Risorse), che ora riporta la nuova istanza registrata.



> **Important**
>
> Devi assicurarti che l'utente e la password utilizzati per registrare l'istanza Amazon RDS corrispondano a un utente e una password validi. In caso contrario, le tue applicazioni non saranno in grado di connettersi all'istanza.

Per registrare indirizzi aggiuntivi, fare clic su Show Unregistered RDS DB instances (Mostra istanze database RDS non registrate) o su + Unregistered RDS DB instances (+ Istanze database RDS non registrate) e ripetere questa procedura. Per ulteriori informazioni su come utilizzare le istanze Amazon RDS con AWS OpsWorks Stacks, consulta. Amazon RDS Service Layer

> **Note**
>
> Puoi anche registrare istanze Amazon RDS tramite la pagina Layers. Per ulteriori informazioni, consulta Amazon RDS Service Layer.

Collegamento e spostamento di risorse

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver registrato una risorsa con uno stack, puoi collegarla a una delle istanze dello stack. Inoltre, puoi spostare una risorsa collegata da un'istanza a un'altra. Tieni presente quanto segue:

- Quando colleghi o sposti volumi Amazon EBS, le istanze coinvolte nell'operazione devono essere offline. Se l'istanza a cui sei interessato non è nella pagina Resources (Risorse), vai alla pagina Instances (Istanze) e arresta l'istanza. Una volta arrestata, puoi tornare alla pagina Resources (Risorse) e collegare o spostare la risorsa.
- Quando colleghi o sposti indirizzi IP elastici, le istanze possono essere online o offline.
- Se elimini un'istanza, le risorse collegate rimangono registrate con lo stack. A questo punto, puoi collegare la risorsa a un'altra istanza o, se non è più necessaria, annullare la registrazione della risorsa.

Argomenti

- Assegnazione di volumi Amazon EBS a un'istanza
- Associazione di indirizzi IP elastici a un'istanza
- Collegamento di istanze Amazon RDS a un'app

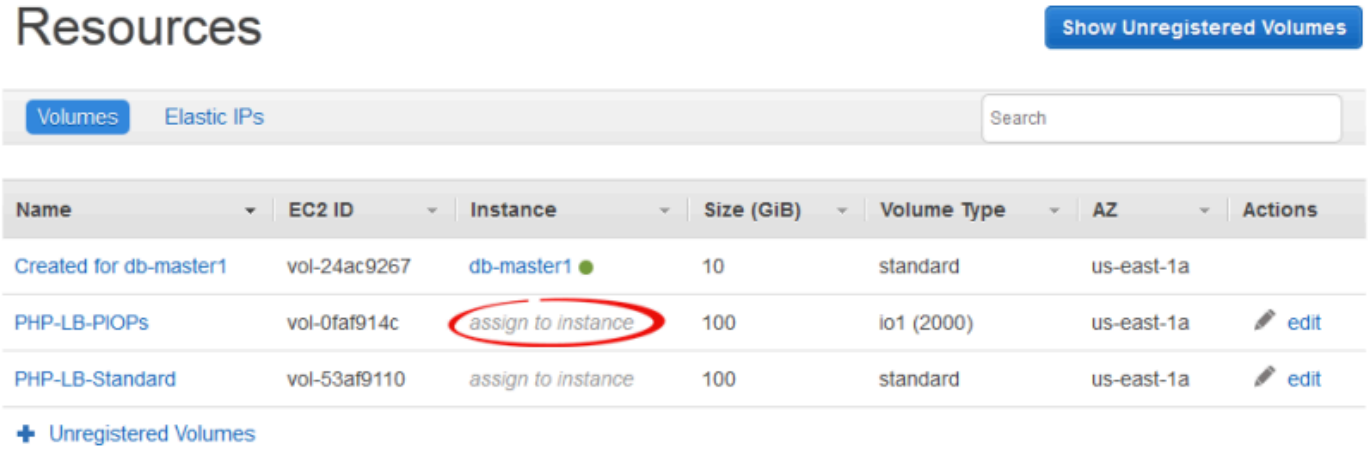
Assegnazione di volumi Amazon EBS a un'istanza

> **Note**
>
> Non è possibile assegnare volumi Amazon EBS a istanze Windows.

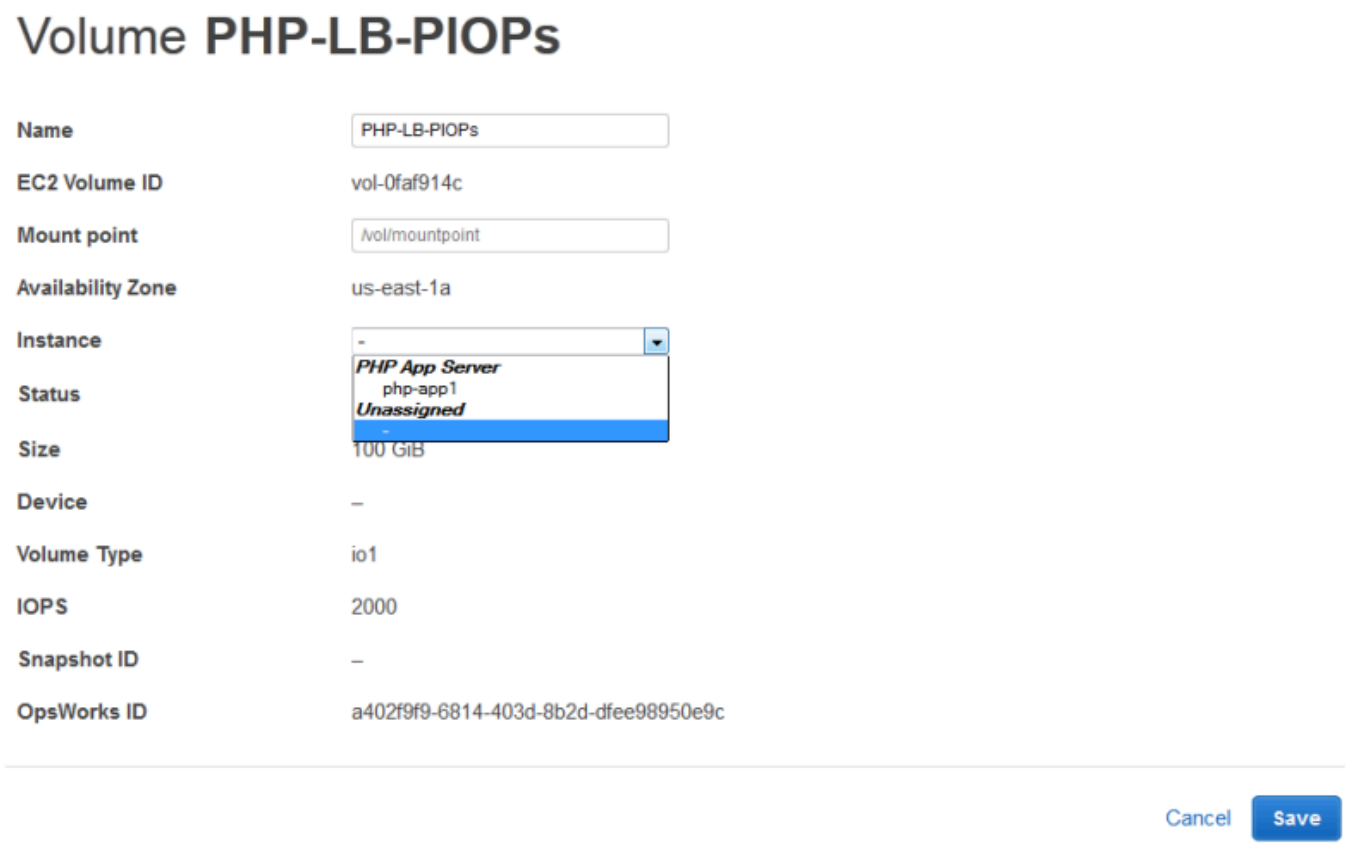
Puoi assegnare un volume Amazon EBS registrato a un'istanza e spostarlo da un'istanza all'altra, ma entrambe le istanze devono essere offline.

Per assegnare un volume Amazon EBS a un'istanza

1. Nella pagina Resources (Risorse) fare clic su assign to instance (assegna a istanza) nella colonna Instance (Istanza) del volume appropriato.



2. Nella pagina dei dettagli del volume, selezionare l'istanza appropriata, specificare il nome del volume e il punto di montaggio, quindi fare clic su Save (Salva) per collegare il volume all'istanza.



> **Important**
>
> Se hai assegnato un volume esterno in uso alla tua istanza, devi utilizzare la EC2 console, l'API o la CLI di Amazon per annullarne l'assegnazione dall'istanza originale o il processo di avvio avrà esito negativo.

Puoi anche utilizzare la pagina dei dettagli per spostare un volume Amazon EBS assegnato su un'altra istanza dello stack.

Per spostare un volume Amazon EBS su un'altra istanza

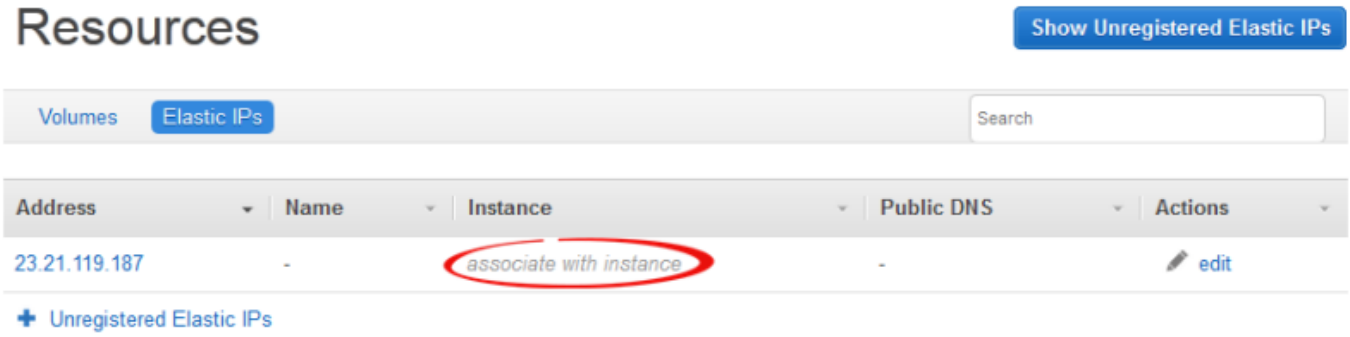
1. Verificare che entrambe le istanze siano in stato offline.
2. Nella pagina Resources (Risorse) fare clic su Volumes (Volumi), quindi su edit (modifica) nella colonna Actions (Azioni) del volume.
3. Esegui una di queste operazioni:
   - Per spostare il volume in un'altra istanza nello stack, selezionare l'istanza appropriata dall'elenco Instance (Istanza) e fare clic su Save (Salva).
   - Per spostare il volume in un'istanza di un altro stack, annullare la registrazione del volume, registrare il volume con il nuovo stack e collegarlo alla nuova istanza.

Associazione di indirizzi IP elastici a un'istanza

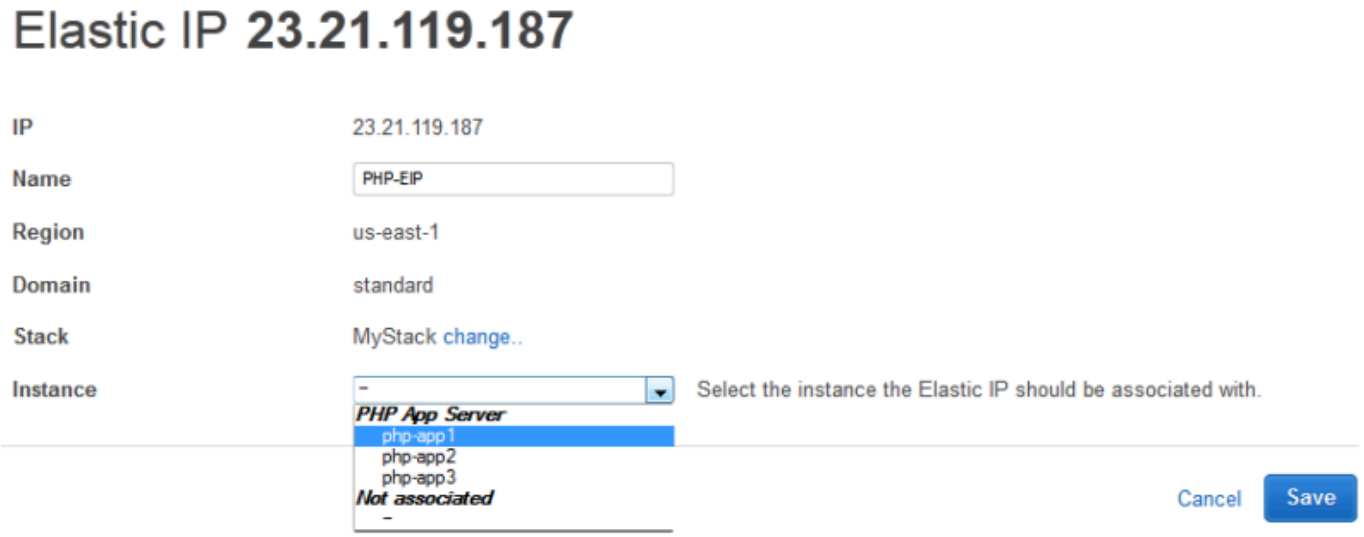
Puoi associare un indirizzo IP elastico registrato a un'istanza e spostarlo da un'istanza a un'altra, incluse le istanze di altri stack. Le istanze possono essere online o offline.

Per associare un indirizzo IP elastico a un'istanza

1. Nella pagina Resources (Risorse) fare clic su associate with instance (associa con istanza) nella colonna Instance (Istanza) dell'indirizzo appropriato.



2. Nella pagina dei dettagli dell'indirizzo, selezionare l'istanza appropriata, specificare il nome dell'indirizzo e fare clic su Save (Salva) per associare l'indirizzo all'istanza.



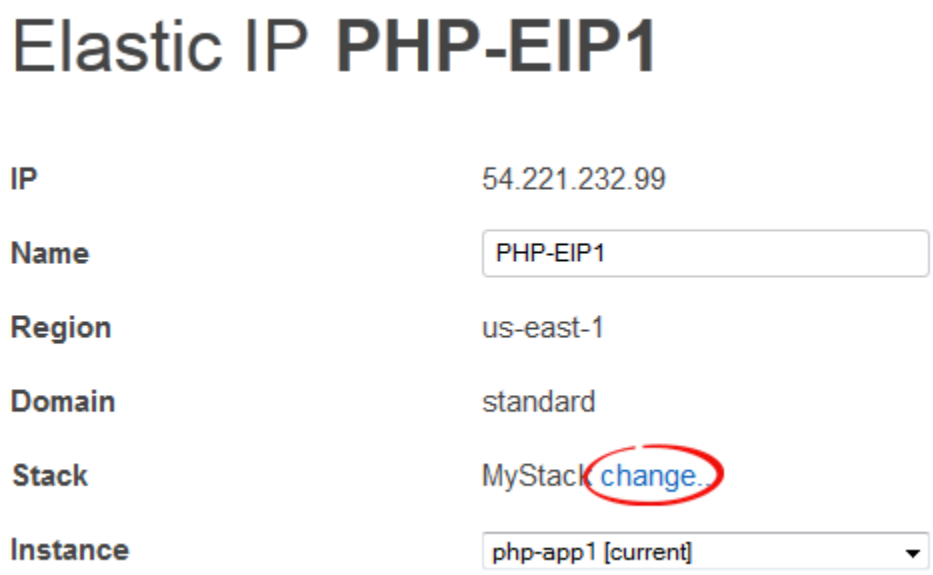
> **Note**
>
> Se l'indirizzo IP elastico è attualmente associato a un'altra istanza online, AWS OpsWorks Stacks riassegna automaticamente l'indirizzo alla nuova istanza.

Puoi utilizzare la pagina dei dettagli per spostare un indirizzo IP elastico associato a un'altra istanza.

Per spostare un indirizzo IP elastico in un'altra istanza

1. Nella pagina Risorse, fai clic su Elastic IPs e fai clic su Modifica nella colonna Azioni dell'indirizzo.
2. Esegui una di queste operazioni:

- Per spostare l'indirizzo in un'altra istanza nello stack, selezionare l'istanza appropriata dall'elenco Instance (Istanza) e fare clic su Save (Salva).
- Per spostare l'indirizzo in un'istanza di un altro stack, fare clic su change (modifica) nelle impostazioni Stack per consultare un elenco di tutti gli stack disponibili. Selezionare uno stack dall'elenco Stack e un'istanza dall'elenco Instance (Istanza). Quindi, fare clic su Save (Salva).



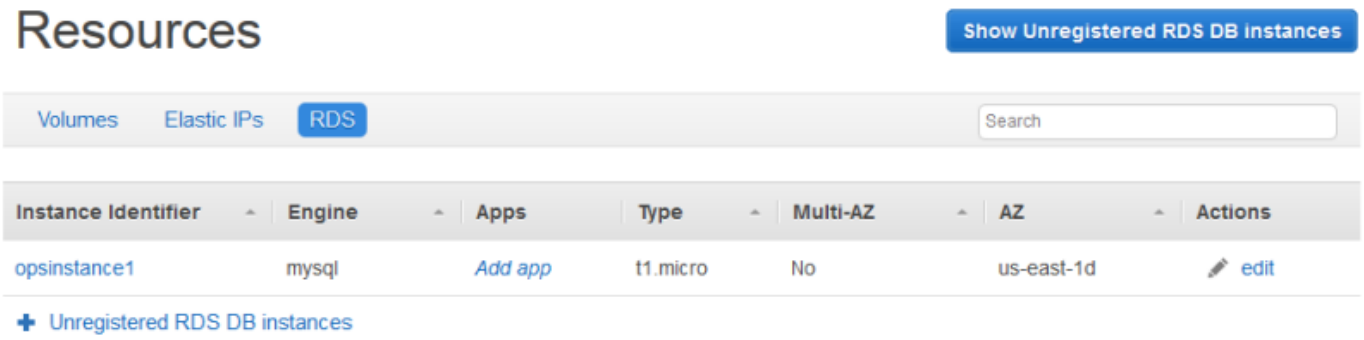
Dopo aver collegato o spostato un indirizzo, AWS OpsWorks Stacks attiva un evento Configure lifecycle per notificare la modifica alle istanze dello stack.

Collegamento di istanze Amazon RDS a un'app

Puoi collegare un'istanza Amazon RDS a una o più app.

Per collegare un'istanza Amazon RDS a un'app

1. Nella pagina Resources (Risorse) fare clic su Add app (Aggiungi app) nella colonna Apps (App) dell'istanza appropriata.



2. Utilizza la pagina Aggiungi app per collegare l'istanza Amazon RDS. Per ulteriori informazioni, consulta Aggiunta di app.

Poiché un Amazon RDS può essere collegato a più app, non esiste una procedura speciale per spostare l'istanza da un'app all'altra. È sufficiente che modifichi la prima app per rimuovere l'istanza RDS o la seconda app per aggiungere l'istanza RDS. Per ulteriori informazioni, consulta Modifica delle app.

Scollegamento delle risorse

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Quando non è più necessaria, una risorsa può essere scollegata. Questa risorsa rimane registrata con lo stack e può essere collegata altrove.

Argomenti

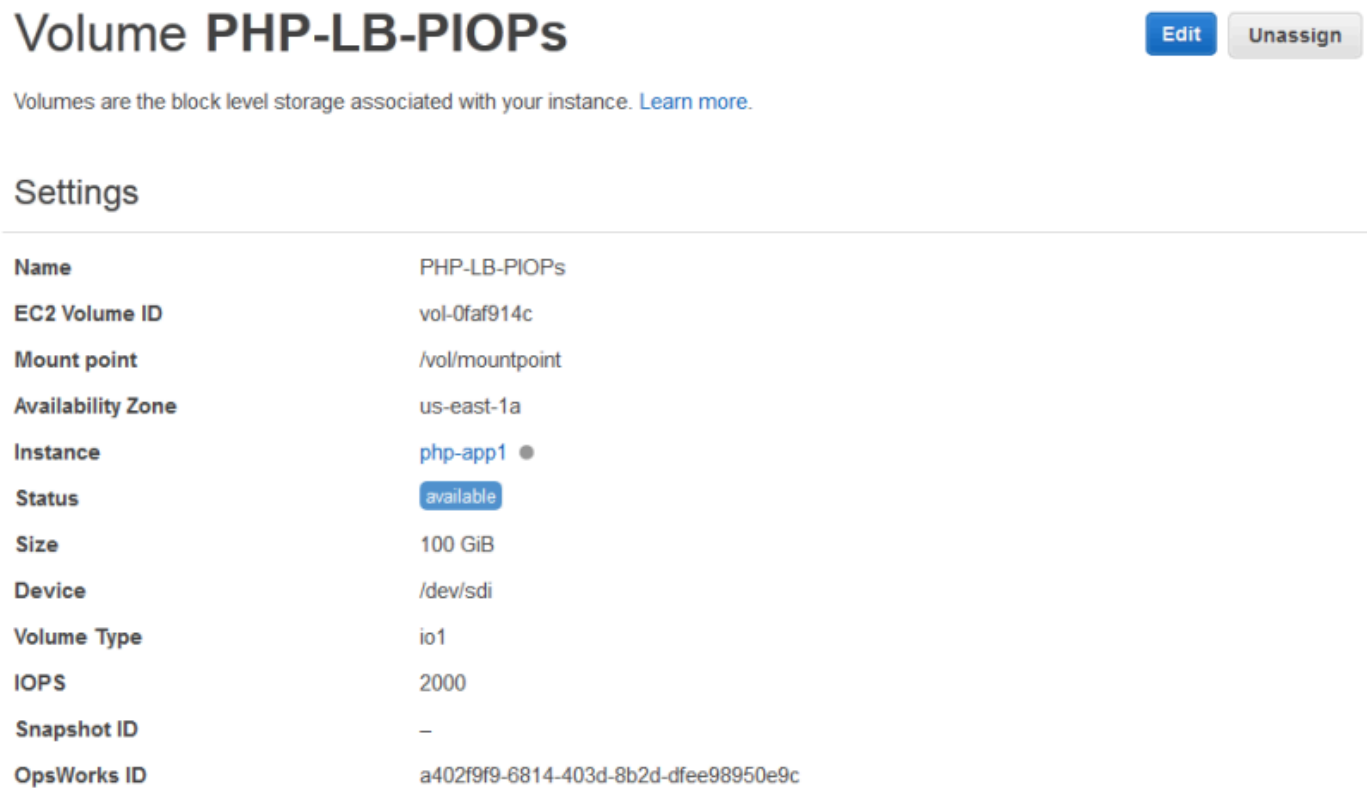
- Annullamento dell'assegnazione di volumi Amazon EBS
- Annullamento dell'associazione di indirizzi IP elastici
- Scollegare le istanze Amazon RDS

Annullamento dell'assegnazione di volumi Amazon EBS

Utilizza la seguente procedura per annullare l'assegnazione di un volume Amazon EBS dalla relativa istanza.

Per annullare l'assegnazione di un volume Amazon EBS

1. Verificare che l'istanza sia in stato offline.
2. Nella pagina Resources (Risorse) fare clic su Volumes (Volumi) e fare clic sul nome del volume.
3. Nella pagina dei dettagli del volume fare clic su Unassign (Annulla assegnazione).



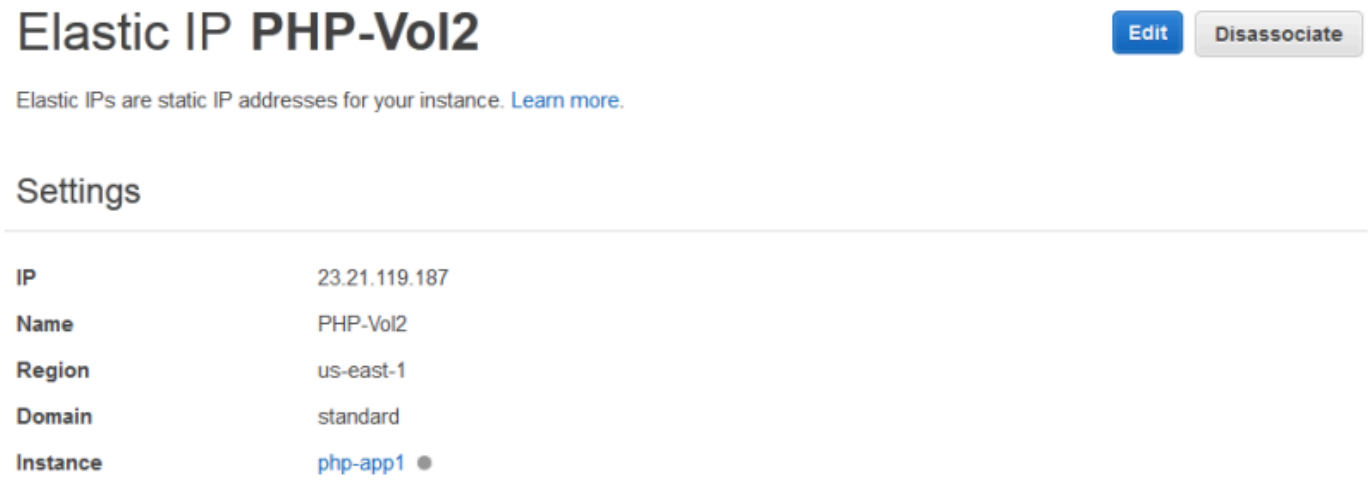
Annullamento dell'associazione di indirizzi IP elastici

Utilizza la procedura seguente per annullare l'associazione di un indirizzo IP elastico alla relativa istanza.

Per annullare l'associazione di un indirizzo IP elastico

1. Nella pagina Risorse, fai clic su Elastic IPs e fai clic su Modifica nella colonna Azioni dell'indirizzo.

2. Nella pagina dei dettagli dell'indirizzo fare clic su Disassociate (Annulla associazione).



Dopo aver dissociato un indirizzo, AWS OpsWorks Stacks attiva un evento Configure lifecycle per notificare la modifica alle istanze dello stack.

Scollegare le istanze Amazon RDS

Utilizza la seguente procedura per scollegare un Amazon RDS da un'app.

Per scollegare un'istanza Amazon RDS

1. Nella pagina Resources (Risorse) fare clic su RDS, quindi fare clic sull'app appropriata nella colonna Apps (App).
2. Fare clic su Edit (Modifica) e modificare la configurazione dell'app per scollegare l'istanza. Per ulteriori informazioni, consulta Modifica delle app.

> **Note**
>
> Questa procedura scollega un Amazon RDS da una singola app. Se l'istanza è collegata a più app, devi ripetere questa procedura per ogni app.

Annullamento della registrazione delle risorse

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se non è più necessario disporre di una risorsa registrata con uno stack, puoi annullarne la registrazione. L'annullamento della registrazione non elimina la risorsa dal tuo account; rimane lì e può essere registrata in un altro stack o utilizzata al di fuori di Stacks. AWS OpsWorks Se desideri eliminare la risorsa interamente, hai a disposizione due opzioni:

- Se un IP elastico o una risorsa Amazon EBS è collegata a un'istanza, puoi eliminare la risorsa quando elimini l'istanza.

  Andare alla pagina Instances (Istanze) e fare clic su delete (elimina) nella colonna Actions (Azioni), quindi selezionare Delete instance's EBS volumes (Elimina volumi EBS dell'istanza) o su Delete the instance's Elastic IP (Elimina IP elastico dell'istanza).
- Annulla la registrazione della risorsa, quindi utilizza la console Amazon o EC2 Amazon RDS, l'API o la CLI per eliminarla.

Argomenti

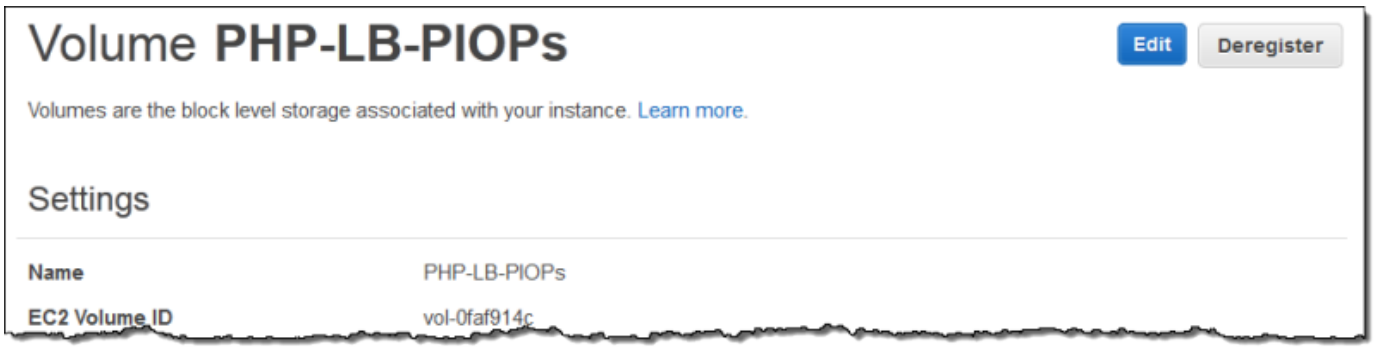
- Annullamento della registrazione dei volumi Amazon EBS
- Annullamento della registrazione di indirizzi IP elastici
- Annullamento della registrazione delle istanze Amazon RDS

Annullamento della registrazione dei volumi Amazon EBS

Utilizza la seguente procedura per annullare la registrazione di un volume Amazon EBS.

Per annullare la registrazione di un volume Amazon EBS

1. Se il volume è collegato a un'istanza, annullare l'assegnazione come descritto in Annullamento dell'assegnazione di volumi Amazon EBS.
2. Nella pagina Resources (Risorse) fare clic sul nome del volume nella colonna Name (Nome).
3. Nella pagina dei dettagli del volume fare clic su Deregister (Annulla registrazione).

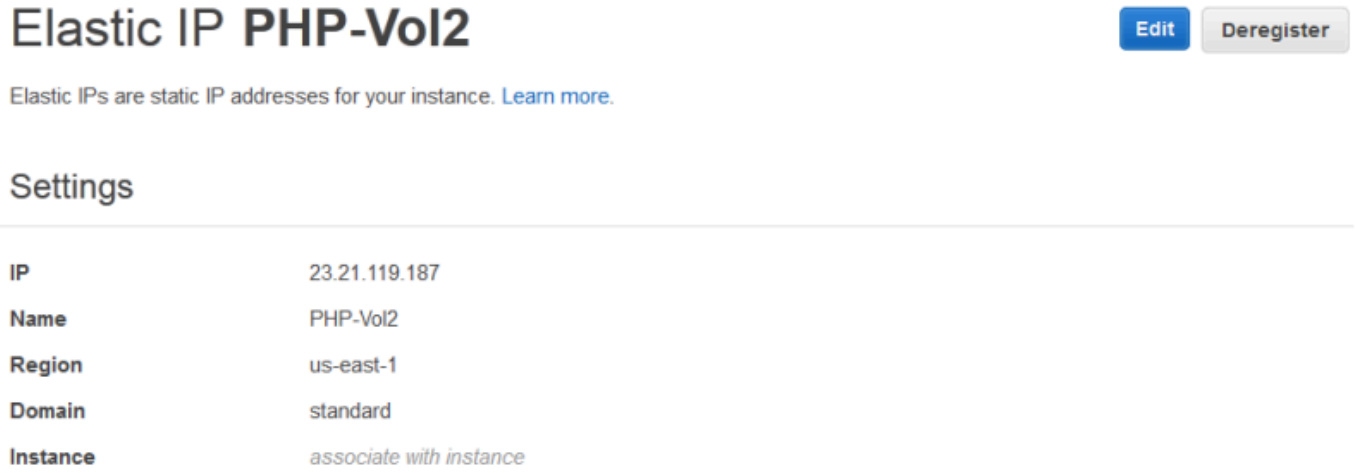


Annullamento della registrazione di indirizzi IP elastici

Utilizza la procedura seguente per annullare la registrazione di un indirizzo IP elastico.

Per annullare la registrazione di un indirizzo IP elastico

1. Se l'indirizzo è associato a un'istanza, annullare l'associazione come descritto in Annullamento dell'associazione di indirizzi IP elastici.
2. Nella pagina Risorse, fai clic su Elastic IPs, quindi sull'indirizzo IP nella colonna Indirizzo.
3. Nella pagina dei dettagli dell'indirizzo fare clic su Deregister (Annulla registrazione).



> **Note**
>
> Se desideri semplicemente registrare un indirizzo IP elastico con un altro stack, devi annullare la registrazione allo stack corrente e registrarlo con il nuovo stack. Tuttavia, puoi spostare direttamente un indirizzo IP elastico collegato in un'istanza di un altro stack. Per ulteriori informazioni, consulta Collegamento e spostamento di risorse.

Annullamento della registrazione delle istanze Amazon RDS

Utilizza la seguente procedura per annullare la registrazione di un'istanza Amazon RDS.

Per annullare la registrazione di un'istanza Amazon RDS

1. Se l'istanza è associata a un'app, scollegarla come descritto in Scollegamento delle risorse.
2. Nella pagina Resources (Risorse) fare clic su RDS, quindi sul nome dell'istanza.
3. Nella pagina dei dettagli dell'istanza fare clic su Deregister (Annulla registrazione).



Tag

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

I tag semplificano il raggruppamento delle risorse in stack Chef 11.10, Chef 12 e Chef 12.2 e la traccia dei costi di utilizzo di tali risorse in AWS Billing and Cost Management.

È possibile applicare tag a livello di stack e livello. Quando crei un tag, lo stai applicando a ogni risorsa all'interno della struttura con tag. Ad esempio, se applichi un tag a un layer, lo applichi a ogni istanza, volume Amazon EBS (tranne la radice) o Elastic Load Balancing load Balancing load Balancer nel layer. I tag non possono essere applicati al volume EBS root o predefinito di un'istanza.

I tag sono coppie chiave-valore che assegnate a pile o livelli in Stacks. AWS OpsWorks Dopo aver creato i tag, apri la console Billing and Cost Management per attivare i tag definiti dall'utente. Per ulteriori informazioni su come attivare i tag e utilizzarli per tracciare e gestire i costi delle risorse AWS OpsWorks Stacks, consulta Utilizzo dei tag di allocazione dei costi e Attivazione dei tag di allocazione dei costi definiti dall'utente nella Guida per l'utente di Billing and Cost Management.

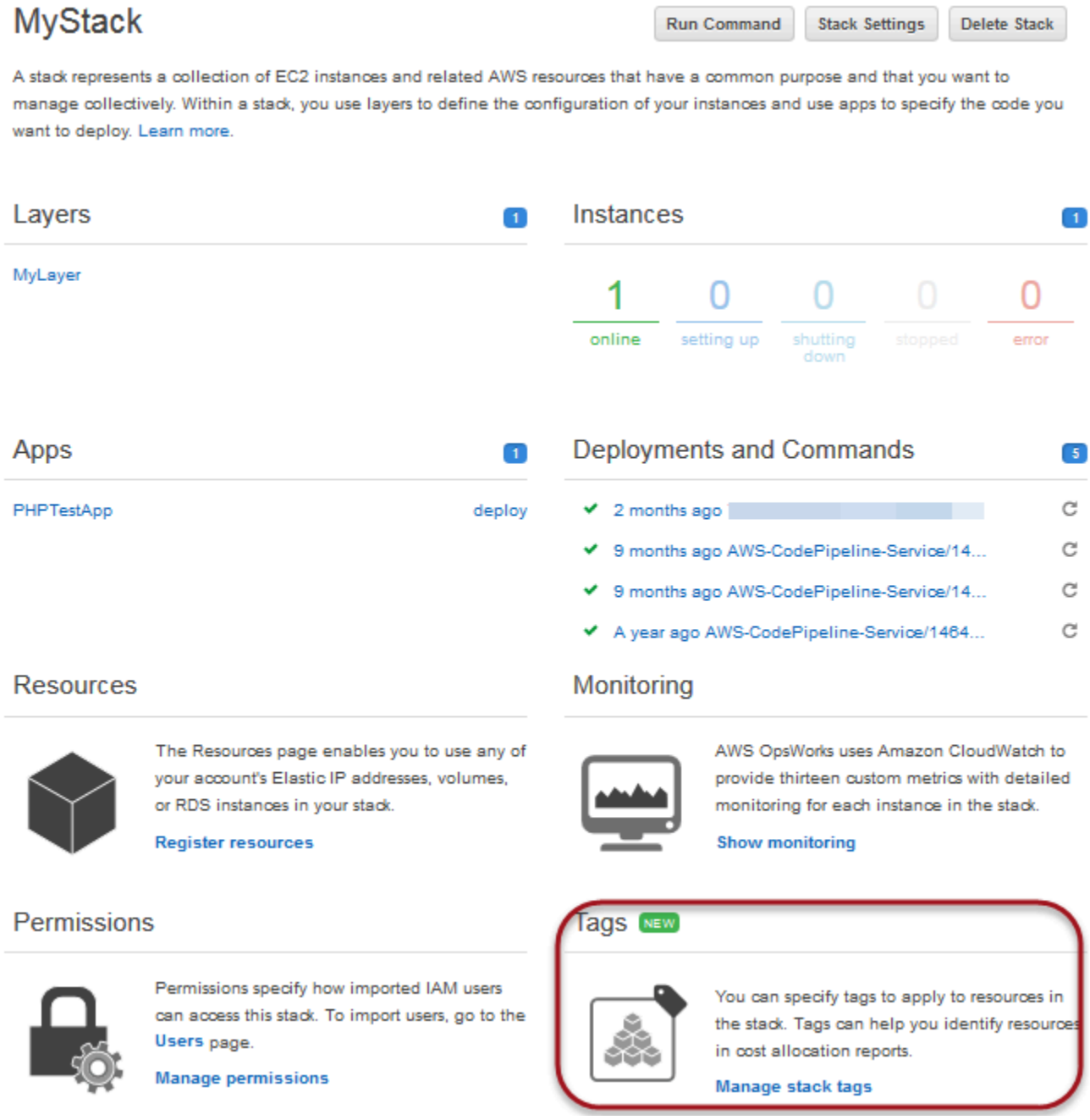
I tag funzionano in modo simile agli attributi personalizzati in Stacks. AWS OpsWorks I tag applicati a uno stack vengono ereditati da ognuno livello nello stack. A livello di livello, puoi sovrascrivere i valori (ma non i nomi delle chiavi) dei tag ereditati e aggiungere nuovi tag specifici del livello. AWS OpsWorks applica il set di tag risultante a tutte le risorse del layer. Mentre crei nuove risorse o assegni risorse esistenti a un livello, le nuove risorse nel livello vengono contrassegnate con lo stesso set di tag.

Argomenti

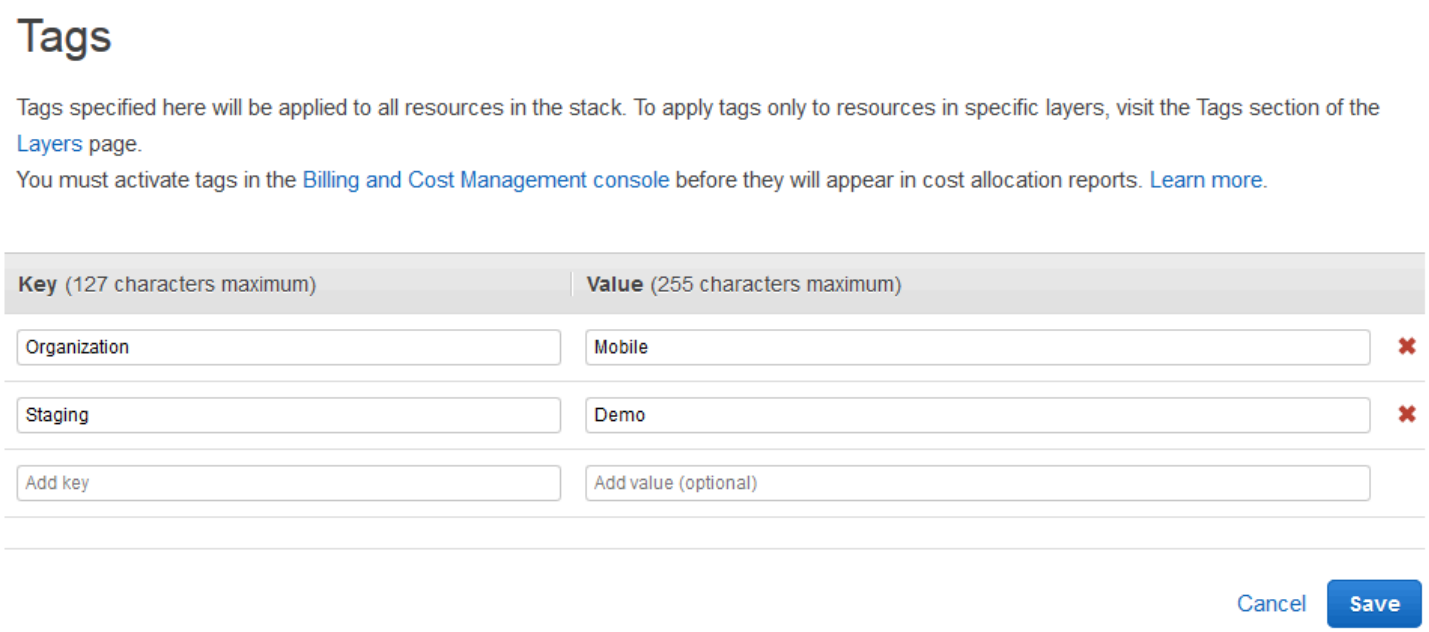
- Impostazione di tag a livello di stack
- Impostazione di tag a livello di livello
- Gestione dei tag con AWS CLI
- Limitazioni dei tag

Impostazione di tag a livello di stack

Al livello di stack, puoi aggiungere e gestire i tag scegliendo Tags (Tag) nella home page dello stack.

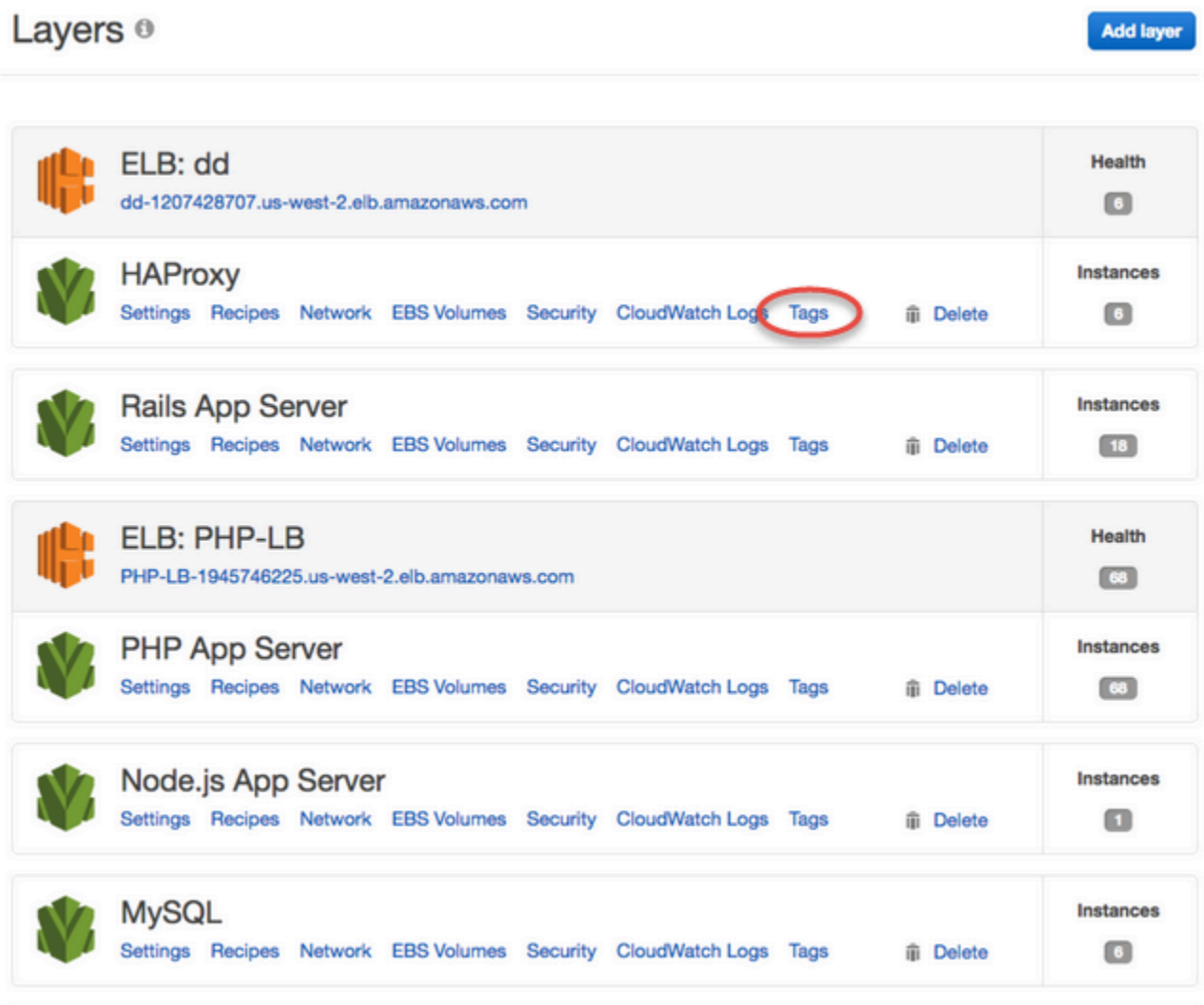


Nella pagina Tags (Tag), aggiungi i tag come coppie chiave-valore. Lo screenshot seguente mostra alcuni tag di esempio. Puoi eliminare i tag scegliendo la X rossa a destra di una coppia chiave-valore.

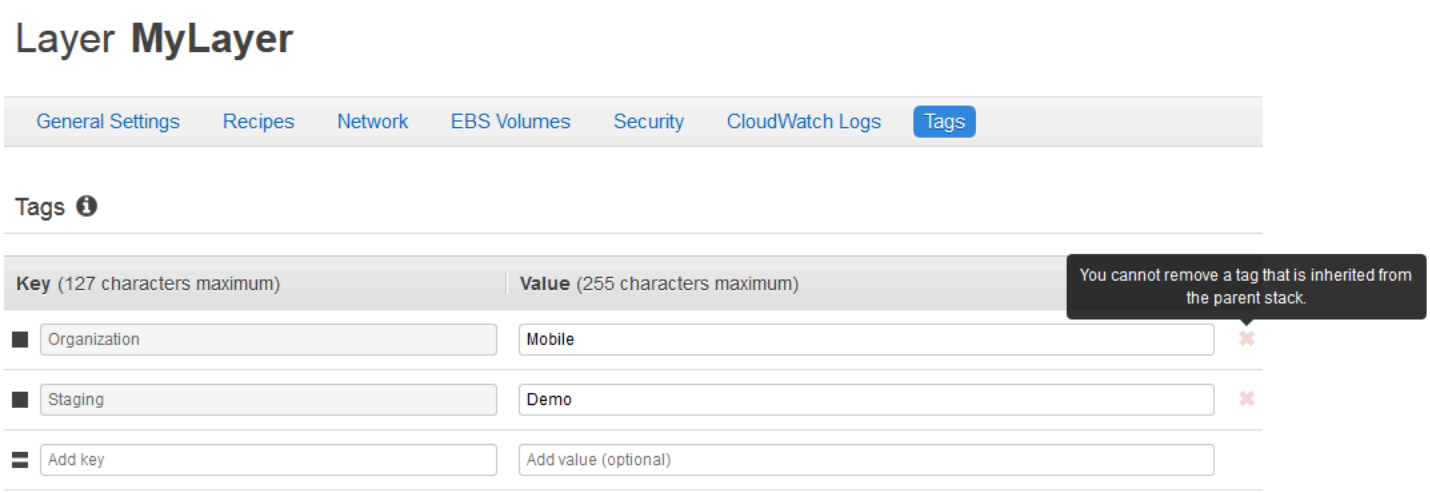


Impostazione di tag a livello di livello

Al livello di livello imposta i tag scegliendo la scheda Tags (Tag). Puoi trovare questa scheda nella home page Layers (Livelli) e nella home page per ogni singolo livello.



Quando modifichi o aggiungi i tag a livello di livello, ricorda che i tag aggiunti al livello di stack padre vengono ereditate dal livello e dalle relative risorse. Mentre puoi modificare i valori dei tag ereditati, non puoi modificare i nomi delle chiavi o eliminare i tag ereditati. Modifica i nomi delle chiavi o elimina i tag ereditati dallo stack padre nelle impostazioni dello stack. Il seguente screenshot mostra esempi di tag ereditato dal livello dello stack. I tag ereditati sono ombreggiati.



Per ulteriori informazioni sull'aggiunta di tag agli stack, consulta Creare un nuovo stack. Per ulteriori informazioni sull'aggiunta di tag ai livelli, consulta Modifica della configurazione di un livello OpsWorks .

Gestione dei tag con AWS CLI

Puoi anche usare AWS CLI i comandi per aggiungere e rimuovere tag a livello di pila e livello. Per ulteriori informazioni sul download e l'installazione di AWS CLI, vedere Installazione dell'interfaccia a riga di AWS comando. Ricorda di aggiungere il parametro `--region` al comando se lo stack che desideri contrassegnare con dei tag non si trova nella regione predefinita. ARNs I livelli non sono attualmente visualizzati in AWS Management Console. Per recuperare l'ARN di un livello, esegui il comando describe-layers.

Per aggiungere tag utilizzando il AWS CLI

- Al prompt dei AWS CLI comandi, digitate il comando seguente, sostituendo ***stack_or_layer_ARN*** e specificando i tag della coppia chiave-valore, quindi premete Invio. Le virgolette doppie vengono precedute da caratteri di escape, ovvero le barre rovesciate.

  ```
  aws opsworks tag-resource --resource-arn stack_or_layer_ARN --tags "{\"key\":
  \"value\",\"key\":\"value\"}"
  ```

  Di seguito è riportato un esempio.

  ```
  aws opsworks tag-resource --resource-arn arn:aws:opsworks:us-
  east-2:800000000003:stack/500b99c0-ec00-4cgg-8a0d-1000000jjd1b --tags "{\"Stage\":
  \"Production\",\"Organization\":\"Mobile\"}"
  ```

Per rimuovere i tag utilizzando il AWS CLI

- Al AWS CLI prompt dei comandi, digitare quanto segue e quindi premere Invio.

```
aws opsworks untag-resource --resource-arn stack_or_layer_ARN --tag-keys "[\"key\",
\"key\"]"
```

Per rimuovere i tag, specificare solo la chiave del tag che si desidera rimuovere. Di seguito è riportato un esempio.

```
aws opsworks untag-resource --resource-arn arn:aws:opsworks:us-
east-2:800000000003:stack/500b99c0-ec00-4cgg-8a0d-1000000jjd1b --tag-keys "[\"Stage
\",\"Organization\"]"
```

> **Note**
>
> Non puoi rimuovere i tag ereditati (i tag aggiunti al livello di stack padre) da un livello. Rimuovi invece i tag ereditati dallo stack.

Limitazioni dei tag

Quando crei i tag, considera le seguenti limitazioni.

- AWS OpsWorks Stacks limita a 40 il numero di tag definiti dall'utente a livello di pila e livello, inclusi i tag definiti dall'utente ereditati da un livello principale. Ciò lascia 10 slot disponibili per i tag predefiniti a cui viene aggiunto e per i tag impostati `opsworks:` da altri processi. AWS È consentito un massimo di 50 tag su una risorsa, inclusi i tag definiti dall'utente e i tag predefiniti creati da. AWS
- Le chiavi dei tag non possono iniziare con **`aws:`**, **`opsworks:`** o **`rds:`**. Non utilizzare **`name`** or **`Name`** come chiave di tag, poiché **`Name`** è riservata da AWS OpsWorks Stacks.
- Una chiave può avere un massimo di 127 caratteri e può contenere solo lettere, numeri o separatori Unicode oppure i seguenti caratteri speciali: `+ - = . _ : /` .
- Un valore può avere un massimo di 255 caratteri e può contenere solo lettere, numeri o separatori Unicode oppure i seguenti caratteri speciali: `+ - = . _ : /` .

Monitoraggio

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

È possibile monitorare gli stack nei modi seguenti.

- AWS OpsWorks Stacks utilizza Amazon CloudWatch per fornire tredici parametri personalizzati con monitoraggio dettagliato per ogni istanza dello stack.
- AWS OpsWorks Stacks si integra con AWS CloudTrail per registrare ogni chiamata API AWS OpsWorks Stacks e archiviare i dati in un bucket Amazon S3.
- Puoi utilizzare Amazon CloudWatch Logs per monitorare il sistema, l'applicazione e i log personalizzati dello stack.

Argomenti

- Monitoraggio degli stack tramite Amazon CloudWatch
- Registrazione delle chiamate API AWS OpsWorks Stacks con AWS CloudTrail
- Utilizzo di Amazon CloudWatch Logs con Stacks AWS OpsWorks
- Monitoraggio degli stack tramite Amazon Events CloudWatch

Monitoraggio degli stack tramite Amazon CloudWatch

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks utilizza Amazon CloudWatch (CloudWatch) per fornire il monitoraggio degli stack.

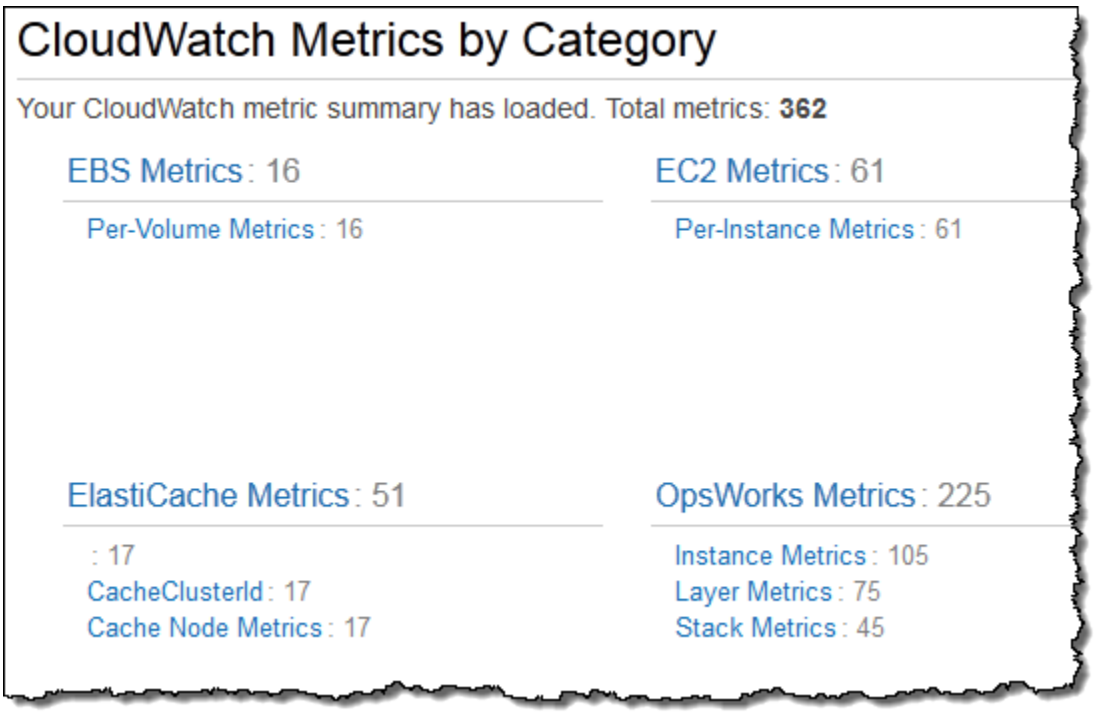
- Per gli stack Linux, AWS OpsWorks Stacks supporta tredici metriche personalizzate per fornire un monitoraggio dettagliato per ogni istanza dello stack e riepiloga i dati per comodità nella pagina Monitoraggio.
- Per gli stack Windows, puoi monitorare i EC2 parametri Amazon standard per le tue istanze con la console. CloudWatch

  La pagina Monitoring (Monitoraggio) non visualizza i parametri Windows.

La pagina Monitoraggio mostra i parametri per un intero stack, un layer o un'istanza. AWS OpsWorks Le metriche degli stack sono diverse dalle metriche di Amazon EC2 . Puoi anche abilitare metriche aggiuntive tramite la CloudWatch console, ma in genere richiedono costi aggiuntivi. Puoi anche visualizzare i dati sottostanti sulla CloudWatch console, come segue:

Per visualizzare le metriche OpsWorks personalizzate in CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nella barra di navigazione selezionare la regione dello stack.
3. Nel riquadro di navigazione, seleziona Parametri.
4. In OpsWorks Metrics, scegli Instance Metrics, Layer Metrics o Stack Metrics.



> **Note**
>
> AWS OpsWorks Stacks raccoglie le metriche eseguendo un processo su ogni istanza (l'agent dell'istanza). Poiché CloudWatch raccoglie le metriche in modo diverso, utilizzando l'hypervisor, i valori nella CloudWatch console potrebbero differire leggermente dai valori corrispondenti nella pagina Monitoring della console Stacks. AWS OpsWorks

Puoi anche usare CloudWatch la console per impostare gli allarmi. Per ulteriori informazioni su come creare allarmi, consulta Creazione di CloudWatch allarmi Amazon. Per un elenco di metriche CloudWatch personalizzate, consulta AWS OpsWorks Metrics and Dimensions. Per ulteriori informazioni, consulta Amazon CloudWatch.

Argomenti

- AWS OpsWorks Stacks Metrics
- Dimensioni per AWS OpsWorks Stacks Metrics
- Parametri stack
- Parametri livello
- Parametri istanza

AWS OpsWorks Stacks Metrics

AWS OpsWorks Stacks invia le seguenti metriche ogni cinque minuti. CloudWatch

Metriche della CPU

Parametro	Descrizione
`cpu_idle`	La percentuale di tempo in cui la CPU è inattiva. Dimensioni valide: le IDs singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o. InstanceId Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna
`cpu_nice`	La percentuale di tempo in cui la CPU gestisce processi con un `nice` valore positivo, che hanno una priorità di pianificazione inferiore. Per ulteriori informazioni su ciò che questo misura, vedete nice (Unix). Dimensioni valide: le IDs singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o. InstanceId Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna
`cpu_steal`	Poiché AWS alloca le risorse della CPU dell'hype rvisor tra un numero crescente di istanze, il carico di virtualizzazione aumenta e può influire sulla frequenza con cui l'hypervisor può eseguire il lavoro richiesto su un'istanza. `cpu_steal` misura la

Parametro	Descrizione
	percentuale di tempo in cui un'istanza attende che l'hypervisor allochi le risorse fisiche della CPU. Dimensioni valide: le IDs singole risorse per le quali vengono visualizzate le metriche: StackId, o. LayerId InstanceId Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna
`cpu_system`	La percentuale di tempo in cui la CPU gestisce le operazioni di sistema. Dimensioni valide: IDs le singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o InstanceId. Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna
`cpu_user`	La percentuale di tempo in cui la CPU gestisce le operazioni dell'utente. Dimensioni valide: IDs le singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o InstanceId. Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna

Parametro	Descrizione
`cpu_waitio`	La percentuale di tempo di attesa della CPU per le operazioni di input/output. Dimensioni valide: le IDs singole risorse per le quali stai visualizzando le metriche: StackId,, o. LayerId InstanceId Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna

Metriche della memoria

Parametro	Descrizione
`memory_buffers`	La quantità di memoria bufferizzata. Dimensioni valide: le IDs singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o. InstanceId Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna
`memory_cached`	La quantità di memoria cache. Dimensioni valide: le IDs singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o. InstanceId Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna

Parametro	Descrizione
`memory_free`	La quantità di memoria libera. Dimensioni valide: IDs le singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o InstanceId. Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna
`memory_swap`	La quantità di spazio di swap. Dimensioni valide: le IDs singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o. InstanceId Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna
`memory_total`	La quantità totale di memoria. Dimensioni valide: IDs le singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o InstanceId. Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna

Parametro	Descrizione
`memory_used`	La quantità di memoria in uso. Dimensioni valide: IDs le singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o InstanceId. Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna

Metriche di caricamento

Parametro	Descrizione
`load_1`	Il carico è stato calcolato in media su una finestra di un minuto. Dimensioni valide: le IDs singole risorse per le quali stai visualizzando le metriche: StackId,, LayerId o. InstanceId Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna
`load_5`	Il carico è stato calcolato in media su una finestra di cinque minuti. Dimensioni valide: le IDs singole risorse per le quali stai visualizzando le metriche: StackId,, LayerId o. InstanceId Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`.

Parametro	Descrizione
	Unità: nessuna
`load_15`	Il carico è stato calcolato in media su una finestra di 15 minuti. Dimensioni valide: le IDs singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o. InstanceId Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna

Metriche di processo

Parametro	Descrizione
`procs`	Il numero di processi attivi. Dimensioni valide: IDs le singole risorse per le quali stai visualizzando le metriche: StackId, LayerId, o InstanceId. Statistiche valide:`Average`,`Minimum`, `MaximumSum`, o`Data Samples`. Unità: nessuna

Dimensioni per AWS OpsWorks Stacks Metrics

AWS OpsWorks Le metriche Stacks utilizzano lo spazio dei nomi AWS OpsWorks Stacks e forniscono metriche per le seguenti dimensioni:

Dimensione	Descrizione
`StackId`	Valori medi per uno stack.
`LayerId`	Valori medi per un livello.
`InstanceId`	Valori medi per un'istanza.

Parametri stack

Per visualizzare un riepilogo delle metriche per un intero stack, seleziona uno stack nella dashboard di AWS OpsWorks Stacks, quindi fai clic su Monitoraggio nel riquadro di navigazione. L'esempio seguente fa riferimento a uno stack con un livello PHP e un livello DB.



Nella visualizzazione dello stack sono visualizzati i grafici di quattro tipi di parametri per ogni livello riferimenti a un periodo di tempo specificato: 1 ora, 8 ore, 24 ore, 1 settimana e 2 settimane. Tieni presente quanto segue:

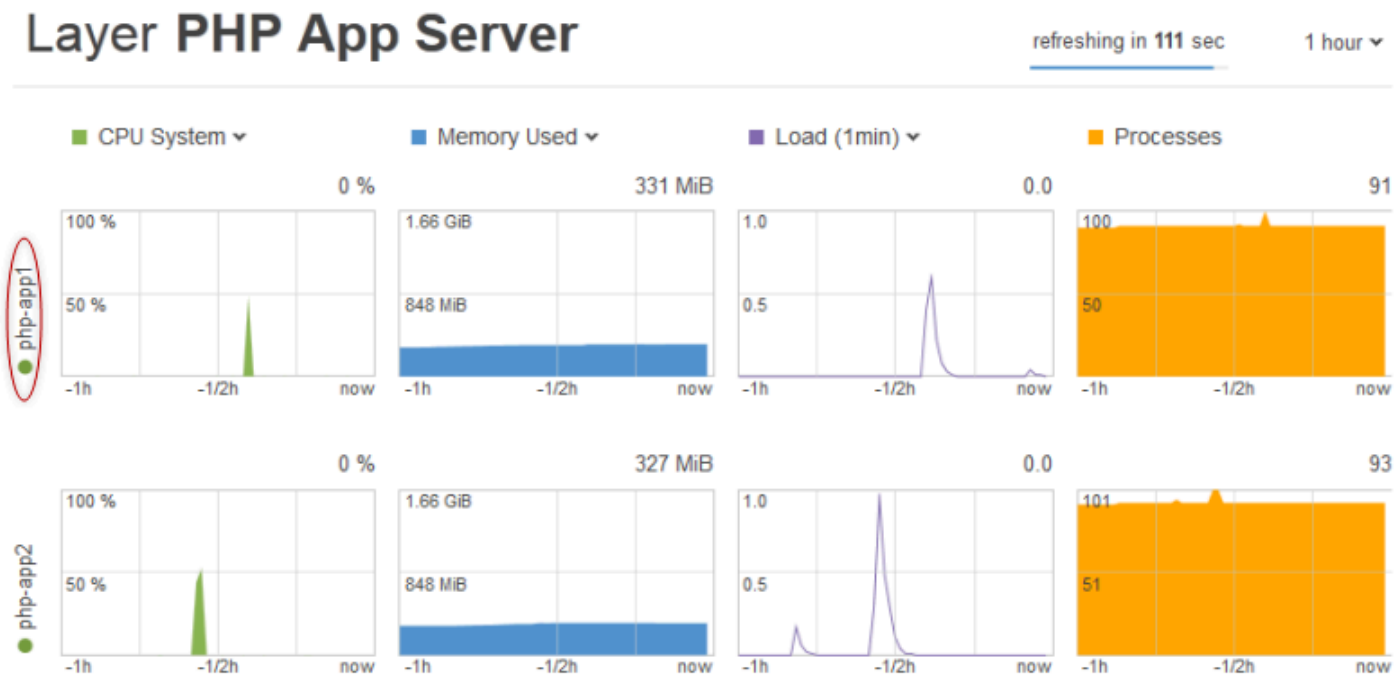
- AWS OpsWorks Stacks aggiorna periodicamente i grafici; il conto alla rovescia in alto a destra indica il tempo rimanente fino al prossimo aggiornamento,
- Se un livello ha più di un'istanza, il grafico visualizza i valori medi per il livello.

- Puoi specificare l'intervallo di tempo facendo clic nell'elenco nell'angolo in alto a destra e selezionando il valore desiderato.

Per ogni tipo di parametro, puoi utilizzare l'elenco nella parte superiore del grafico per selezionare un parametro specifico da visualizzare.

Parametri livello

Per visualizzare i parametri per un determinato livello, fare clic sul nome del livello nella visualizzazione Monitoring Layers (Monitoraggio livelli). L'esempio seguente mostra i parametri per il livello PHP, che ha due istanze.

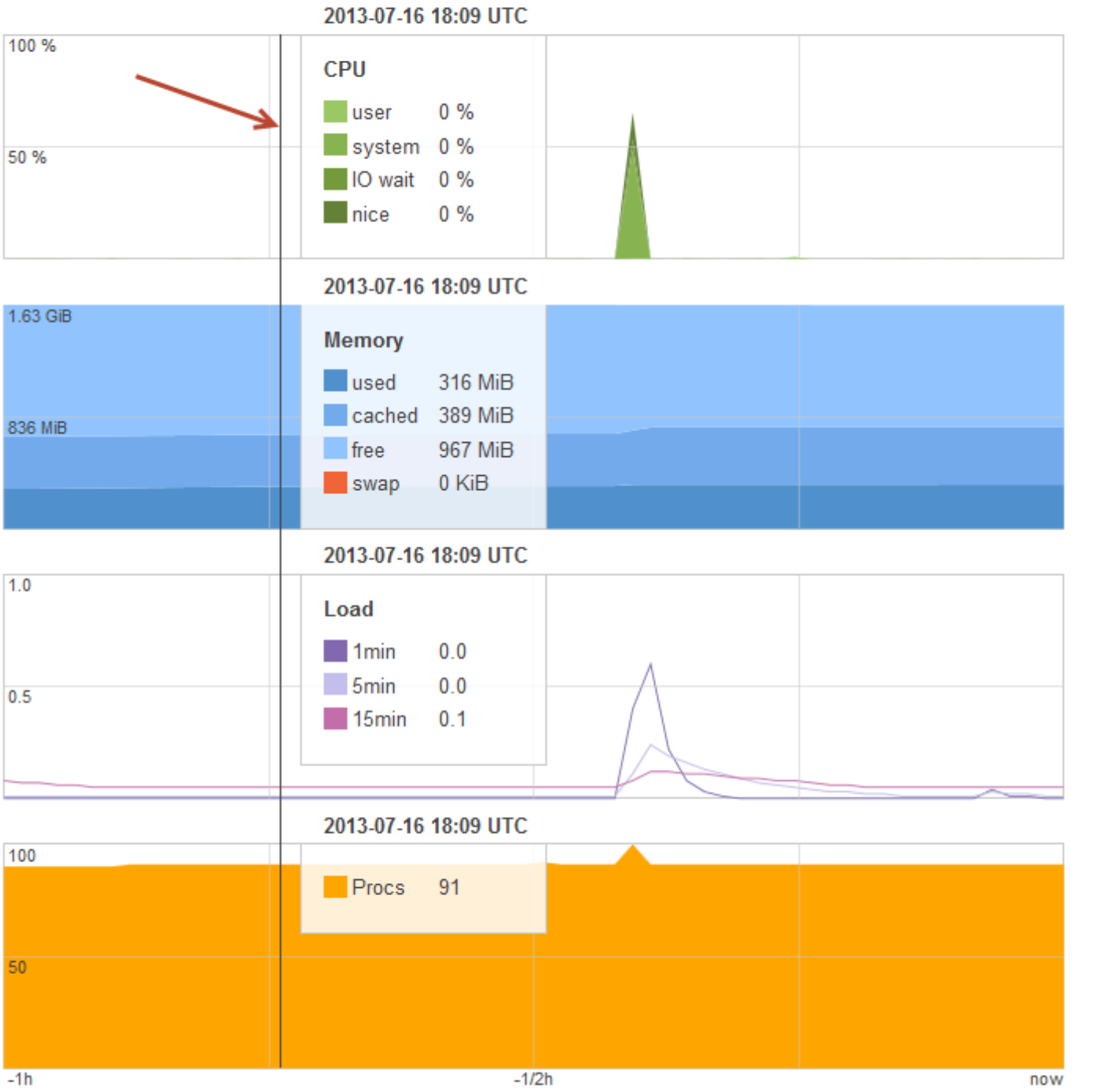


I tipi di parametro sono gli stessi tipi validi per i parametri stack e per ogni tipo puoi utilizzare l'elenco nella parte superiore del grafico per selezionare un parametro specifico da visualizzare.

> **Note**
>
> Puoi anche visualizzare i parametri livello passando alla pagina dei dettagli del livello e facendo clic su Monitoring (Monitoraggio) nell'angolo in alto a destra.

Parametri istanza

Per visualizzare i parametri per una determinata istanza, fare clic sul nome dell'istanza nella visualizzazione Monitoring Layers (Monitoraggio livelli). L'esempio seguente mostra i parametri dell'istanza php-app1 del livello PHP.


Instance php-app1
refreshing in
2013-07-16 18:09 UTC
100 %
50 %
CPU
user 0 %
system 0 %
IO wait 0 %
nice 0 %
2013-07-16 18:09 UTC
1.63 GiB
836 MiB
Memory
used 316 MiB
cached 389 MiB
free 967 MiB
swap 0 KiB
2013-07-16 18:09 UTC
1.0
0.5
Load
1min 0.0
5min 0.0
15min 0.1
2013-07-16 18:09 UTC
100
50
Procs 91
-1h
-1/2h
now

I grafici riepilogano tutti i parametri disponibili per ogni tipo di parametro. Per ottenere i valori esatti per un determinato point-in-time, utilizza il mouse per spostare il cursore (indicata dalla freccia rossa nella figura precedente) nella posizione appropriata.

> **Note**
>
> Puoi anche visualizzare i parametri dell'istanza passando alla pagina dei dettagli dell'istanza e scegliendo Monitoring (Monitoraggio) nell'angolo in alto a destra.

Registrazione delle chiamate API AWS OpsWorks Stacks con AWS CloudTrail

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un'identità IAM o un AWS servizio in Stacks. AWS OpsWorks CloudTrail acquisisce tutte le chiamate API per AWS OpsWorks Stacks come eventi, incluse le chiamate dalla console AWS OpsWorks Stacks e le chiamate di codice agli Stacks. AWS OpsWorks APIs Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Stacks. AWS OpsWorks Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata fatta a AWS OpsWorks Stacks, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida per l'AWS CloudTrail utente.

AWS OpsWorks Accumula le informazioni in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in AWS OpsWorks Stacks, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per AWS OpsWorks Stacks, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un trail nella console, il trail sarà valido in tutte le regioni. Il trail registra gli eventi da tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consultare:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte le azioni di AWS OpsWorks Stacks vengono registrate CloudTrail e documentate nello AWS OpsWorks Stacks API Reference. Ad esempio, le chiamate alle `CreateLayer` `StartInstance` azioni e generano voci nei file di registro. `DescribeInstances` CloudTrail

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali dell'utente IAM o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Comprensione delle AWS OpsWorks voci dei file di registro di Stacks

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'`CreateLayer`azione.


```
  {
"Records": [
    {
        "awsRegion": "us-west-2",
        "eventID": "342cd1ec-8214-4a0f-a68f-8e6352feb5af",
        "eventName": "CreateLayer",
        "eventSource": "opsworks.amazonaws.com",
        "eventTime": "2014-05-28T16:05:29Z",
        "eventVersion": "1.01"ed,
        "requestID": "e3952a2b-e681-11e3-aa71-81092480ee2e",
        "requestParameters": {
            "attributes": {},
            "customRecipes": {},
            "name": "2014-05-28 16:05:29 +0000 a073",
            "shortname": "customcf4571d5c0d6",
            "stackId": "a263312e-f937-4949-a91f-f32b6b641b2c",
            "type": "custom"
        },
        "responseElements": null,
        "sourceIPAddress": "198.51.100.0",
        "userAgent": "aws-sdk-ruby/2.0.0 ruby/2.1 x86_64-linux",
        "userIdentity": {
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "accountId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/A-User-Name",
            "principalId": "AKIAI44QH8DHBEXAMPLE",
            "type": "IAMUser",
            "userName": "A-User-Name"
        }
    },
    {
```

```
            "awsRegion": "us-west-2",
            "eventID": "a860d8f8-c1eb-449b-8f55-eafc373b49a4",
            "eventName": "DescribeInstances",
            "eventSource": "opsworks.amazonaws.com",
            "eventTime": "2014-05-28T16:05:31Z",
            "eventVersion": "1.01",
            "requestID": "e4691bfd-e681-11e3-aa71-81092480ee2e",
            "requestParameters": {
                "instanceIds": [
                    "218289c4-0492-473d-a990-3fbe1efa25f6"
                ]
            },
            "responseElements": null,
            "sourceIPAddress": "198.51.100.0",
            "userAgent": "aws-sdk-ruby/2.0.0 ruby/2.1x86_64-linux",
            "userIdentity": {
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "accountId": "111122223333",
                "arn": "arn:aws:iam::111122223333:user/A-User-Name",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "type": "IAMUser",
                "userName": "A-User-Name"
            }
        }
    ]
}
```


Utilizzo di Amazon CloudWatch Logs con Stacks AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per semplificare il processo di monitoraggio dei log su più istanze, AWS OpsWorks Stacks supporta Amazon Logs. CloudWatch Puoi abilitare CloudWatch i log a livello di livello in Stacks. AWS OpsWorks CloudWatch L'integrazione dei log funziona con gli stack basati su Chef 11.10 e

Chef 12 Linux. Quando abiliti i CloudWatch log, devi sostenere costi aggiuntivi, quindi consulta i CloudWatchprezzi di Amazon prima di iniziare.

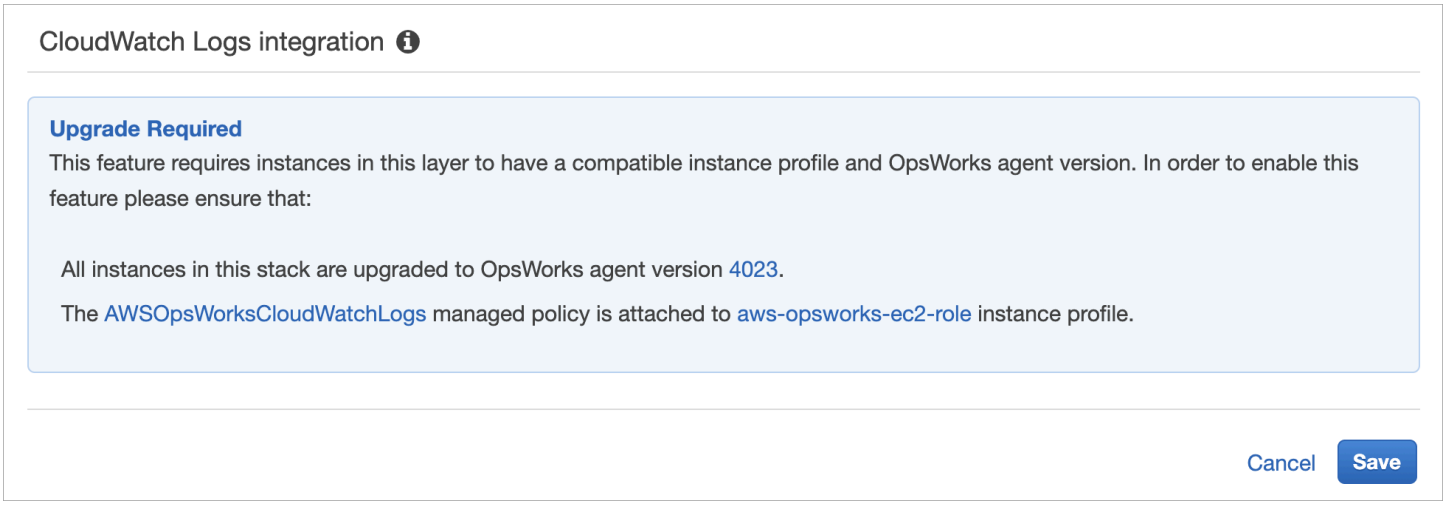
CloudWatch Logs monitora i log selezionati per rilevare la presenza di uno schema specificato dall'utente. Ad esempio, puoi monitorare i log a livello di occorrenza di un termine specifico, ad esempio `NullReferenceException`, di conteggio del numero di tali occorrenze. Dopo aver abilitato CloudWatch Logs in AWS OpsWorks Stacks, l'agente Stacks invia i AWS OpsWorks log a Logs. CloudWatch Per ulteriori informazioni sui CloudWatch log, consulta Getting Started with Logs. CloudWatch

Prerequisiti

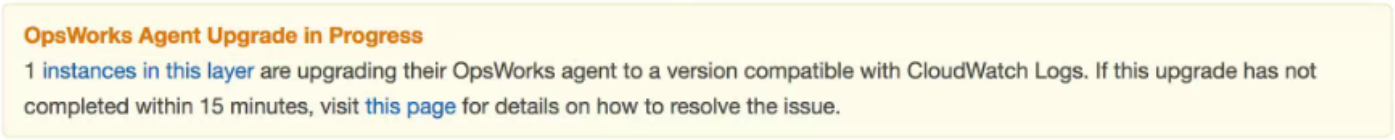
Prima di poter abilitare CloudWatch i log, sulle istanze devono essere in esecuzione la versione 3444 o successiva dell'agente Stacks in Chef 11.10 AWS OpsWorks stacks e 4023 o successiva in Chef 12 stacks. È inoltre necessario utilizzare un profilo di istanza compatibile per tutte le istanze che si stanno monitorando utilizzando Logs. CloudWatch

Se utilizzi un profilo di istanza personalizzato (uno che AWS OpsWorks Stacks non ha fornito al momento della creazione dello stack), AWS OpsWorks Stacks non può aggiornare automaticamente il profilo dell'istanza. È necessario collegare manualmente la AWSOpsWorksCloudWatchLogspolicy al proprio profilo utilizzando IAM. Per informazioni, consulta Managing IAM policy nella IAM User Guide.

Se devi aggiornare la versione dell'agente o il profilo dell'istanza, AWS OpsWorks Stacks visualizza un promemoria simile alla seguente schermata quando apri la scheda CloudWatch Logs nella pagina Layer.



L'aggiornamento dell'agente su tutte le istanze in un livello può richiedere tempo. Se si tenta di abilitare CloudWatch i log su un layer prima che l'aggiornamento dell'agente sia completato, viene visualizzato un messaggio simile al seguente.



Abilitazione dei log CloudWatch

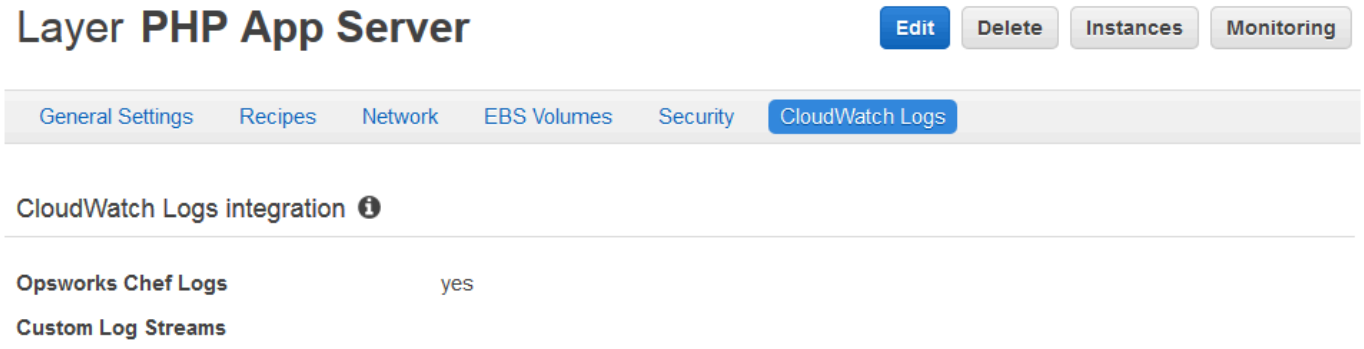
1. Una volta completati gli aggiornamenti necessari del profilo dell'agente e dell'istanza, puoi abilitare CloudWatch i log impostando il cursore nella scheda CloudWatch Registri su Attivato.



2. Per eseguire lo streaming dei log dei comandi, impostare il cursore Stream command logs (Streaming log comandi) su On (Abilitato). Questo invia a Logs i registri delle attività di Chef e dei comandi avviati dall'utente sulle istanze del layer. CloudWatch

   I dati inclusi in questi log corrispondono molto da vicino a quelli che vedete nei risultati di un'DescribeCommandsoperazione, quando aprite la destinazione dell'URL del log. Sono inclusi i dati relativi a `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop` e ai comandi di esecuzione delle ricette.

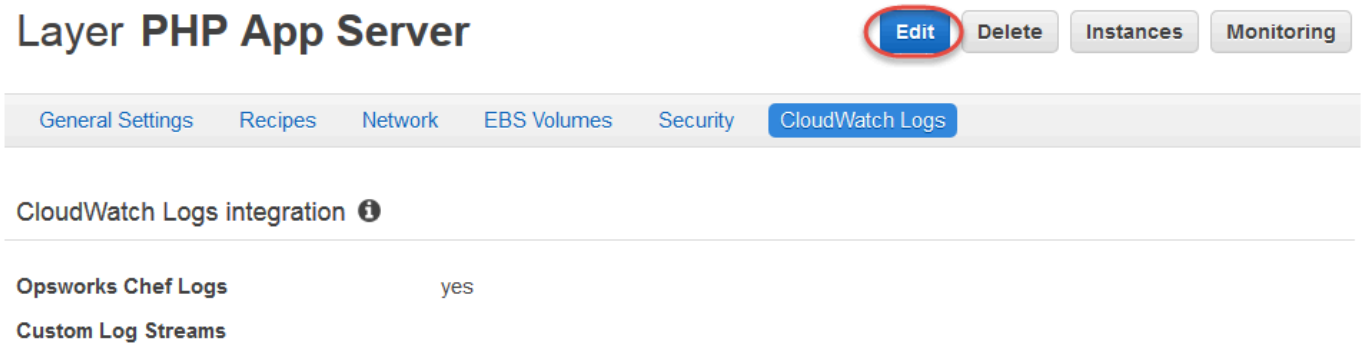
3. Per eseguire lo streaming delle attività archiviate in una posizione personalizzata sulle istanze del livello, ad esempio `/var/log/apache/myapp/mylog*`, immettere la posizione personalizzata nella casella di testo Stream custom logs (Streaming log personalizzati), quindi scegliere Add (Aggiungi) (+).
4. Seleziona Salva. Entro pochi minuti, i flussi di log di AWS OpsWorks Stacks dovrebbero essere visibili nella console Logs. CloudWatch



Disattivazione CloudWatch dei registri

Per disattivare CloudWatch i log, modifica le impostazioni del layer.

1. Nella pagina delle proprietà del livello, scegliere Edit (Modifica).



2. Nella pagina di modifica, scegli la scheda CloudWatch Registri.
3. Nell'area Registri, disattiva CloudWatch i registri dei comandi Stream. Scegliere X sui log personalizzati per eliminarli dai flussi di log, se applicabile.
4. Seleziona Salva.

Eliminazione dei registri in streaming dai registri CloudWatch

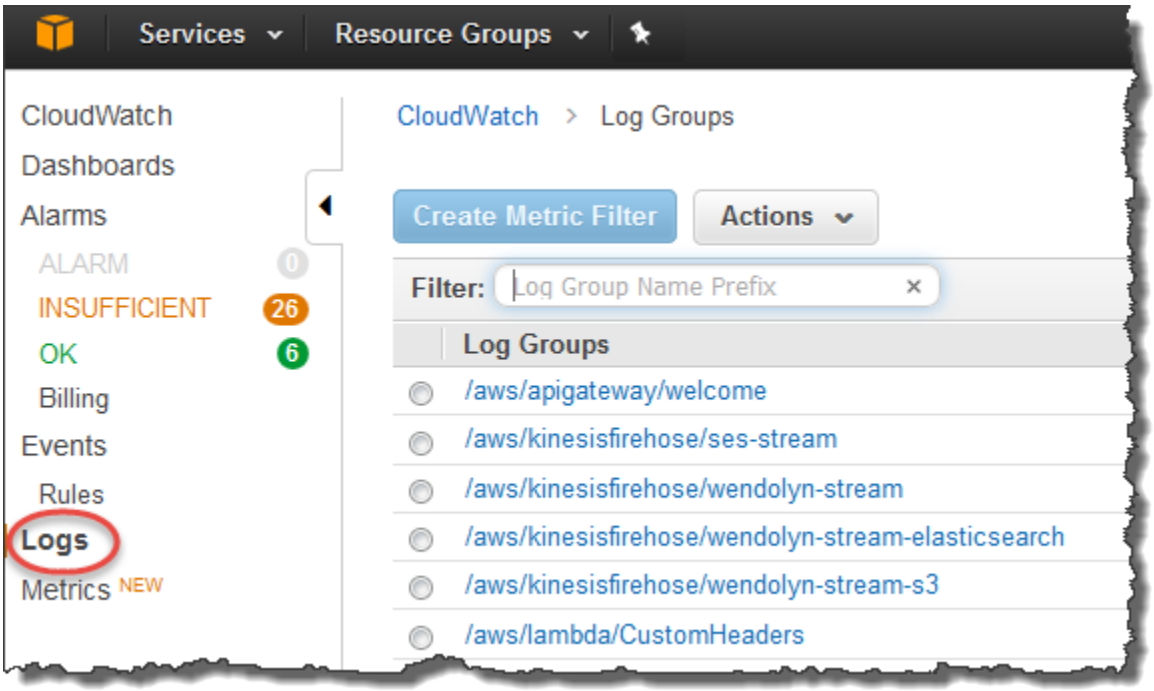
Dopo aver disattivato lo streaming CloudWatch dei log da AWS OpsWorks Stacks, i log esistenti sono ancora disponibili nella console di gestione dei registri. CloudWatch I log archiviati sono comunque soggetti a costi aggiuntivi, a meno che non vengano esportati in Amazon S3 o eliminati. Per ulteriori informazioni sull'esportazione dei log in S3, consulta Esportazione dei dati di log in Amazon S3.

Puoi eliminare i flussi di log e i gruppi di log nella console di gestione dei CloudWatch log o eseguendo i comandi and. `delete-log-stream` `delete-log-group` AWS CLI Per ulteriori

informazioni sulla modifica dei periodi di conservazione dei log, consulta Change Log Data Retention in CloudWatch Logs.

Gestione dei log nei log CloudWatch

I log che stai trasmettendo in streaming vengono gestiti nella console Logs. CloudWatch



AWS OpsWorks crea automaticamente gruppi di log e flussi di log predefiniti. I gruppi di log per i dati di AWS OpsWorks Stacks sono associati a nomi conformi al formato seguente:

stack_name/*layer_name*/*chef_log_name*

I log personalizzati sono associati a nomi conformi al seguente formato:

/stack_name/layer_short_name/file_path_name. Il nome del percorso è reso più leggibile dall'uomo grazie alla rimozione di caratteri speciali, come gli asterischi (*).

Dopo aver individuato i log in CloudWatch Logs, puoi organizzarli in gruppi, cercare e filtrare i logcreando filtri metrici e creare allarmi personalizzati.

Configurazione dei livelli Windows di Chef 12.2 per l'utilizzo dei registri CloudWatch

CloudWatch L'integrazione automatica dei log non è supportata per le istanze basate su Windows. La scheda CloudWatch Registri non è disponibile sui livelli negli stack di Chef 12.2. Per abilitare

manualmente lo streaming su CloudWatch Logs per le istanze basate su Windows, procedi come segue.

- Aggiorna il profilo dell'istanza per le istanze basate su Windows in modo che l'agente Logs disponga delle autorizzazioni appropriate. CloudWatch L'AWSOpsWorksCloudWatchLogsinformativa sulla politica mostra quali autorizzazioni sono richieste.

  In genere, questa attività viene eseguita solo una volta. Potrai quindi utilizzare il profilo dell'istanza aggiornato per tutte le istanze Windows in un livello.
- Modifica il seguente file di configurazione JSON in ciascuna istanza. Questo file include le preferenze di streaming dei log, ad esempio i log da sottoporre a monitoraggio.

  ```
  %PROGRAMFILES%\Amazon\Ec2ConfigService\Settings
  \AWS.EC2.Windows.CloudWatch.json
  ```

Puoi automatizzare le attività precedenti mediante la creazione di ricette personalizzate per gestire le attività richieste e mediante la successiva assegnazione di tali ricette agli eventi Setup (Configurazione) del livello Chef 12.2. Ogni volta che avvii una nuova istanza su questi livelli, AWS OpsWorks Stacks esegue automaticamente le tue ricette al termine dell'avvio dell'istanza, abilitando i log. CloudWatch

Per disattivare CloudWatch i log sulle istanze basate su Windows, inverti la procedura. Deseleziona la casella di controllo Abilita l'integrazione dei CloudWatch registri nella finestra di dialogo Proprietà del EC2 servizio, elimina le preferenze del flusso di registro dal `AWS.EC2.Windows.CloudWatch.json` file e interrompi l'esecuzione di qualsiasi ricetta Chef che assegna automaticamente le autorizzazioni di CloudWatch Logs a nuove istanze nei livelli Chef 12.2.

Monitoraggio degli stack tramite Amazon Events CloudWatch

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi configurare le regole in Amazon CloudWatch Events per avvisarti delle modifiche nelle risorse AWS OpsWorks Stacks e indirizzare CloudWatch Events a intraprendere azioni in base ai contenuti degli eventi. Per ulteriori informazioni su come iniziare a usare CloudWatch Events e configurare le regole, consulta Getting Started with CloudWatch Events nella CloudWatch Events User Guide.

I seguenti tipi di eventi AWS OpsWorks Stacks sono supportati in CloudWatch Events.

Instance state change (Cambio stato istanza)

Indica un cambiamento nello stato di un' AWS OpsWorks istanza Stacks.

Command state change (Cambio stato comando)

Indica che si è verificata una modifica nello stato di un comando AWS OpsWorks Stacks.

Deployment state change (Cambio stato distribuzione)

Indica che si è verificata una modifica nello stato di una distribuzione di AWS OpsWorks Stacks.

Avvisi

Indica che è stato generato un errore del servizio AWS OpsWorks Stacks.

Per ulteriori informazioni sui tipi di eventi AWS OpsWorks Stacks supportati da CloudWatch Events, consulta AWS OpsWorks Stacks Events nella Events User GuideCloudWatch .

Sicurezza e autorizzazioni

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ciascuno dei tuoi utenti deve disporre delle AWS credenziali appropriate per accedere alle risorse del tuo account. AWS Il metodo consigliato per fornire le credenziali agli utenti è con AWS Identity and Access Management(IAM). AWS OpsWorks Stacks si integra con IAM per consentirti di controllare quanto segue:

- In che modo i singoli utenti possono interagire con AWS OpsWorks Stacks.

  Ad esempio, puoi consentire ad alcuni utenti di distribuire le app su qualsiasi stack, ma non modificare lo stack stesso e consentire ad altri utenti l'accesso completo solo a stack specifici e così via.
- In che modo AWS OpsWorks Stacks può agire per tuo conto per accedere a risorse dello stack come EC2 istanze Amazon e bucket Amazon S3.

  AWS OpsWorks Stacks fornisce un ruolo di servizio che concede le autorizzazioni per queste attività.
- In che modo le app eseguite su EC2 istanze Amazon controllate da AWS OpsWorks Stacks possono accedere ad altre AWS risorse, come i dati archiviati nei bucket Amazon S3.

  Puoi assegnare un profilo di istanza alle istanze di un layer che concede le autorizzazioni alle app in esecuzione su tali istanze per accedere ad altre risorse. AWS
- Modalità di gestione delle chiavi SSH basate sugli utenti e modalità di utilizzo di SSH o RDP per connettersi alle istanze.

  Per ogni stack, gli utenti amministratori possono assegnare a ciascun utente una chiave SSH personale o autorizzare gli utenti a specificare le proprie chiavi. Puoi anche autorizzare l'accesso RDP o SSH e concedere i privilegi sudo o amministratore alle istanze dello stack per ogni utente.

Altri aspetti della sicurezza fanno riferimento a quanto segue:

- Modalità di gestione degli aggiornamenti del sistema operativo delle istanze in base alle patch di sicurezza più recenti.

  Per ulteriori informazioni, consulta Gestione degli aggiornamenti della sicurezza.
- Come configurare i gruppi EC2 di sicurezza Amazon per controllare il traffico di rete da e verso le tue istanze.

  Come specificare gruppi di sicurezza personalizzati anziché i gruppi di sicurezza predefiniti di AWS OpsWorks Stack. Per ulteriori informazioni, consulta Utilizzo dei gruppi di sicurezza.

Argomenti

- Gestione delle autorizzazioni utente di AWS OpsWorks Stacks
- Consentire a AWS OpsWorks Stacks di agire per tuo conto

- Prevenzione intersettoriale degli agenti confusi in Stacks AWS OpsWorks
- Specificazione delle autorizzazioni per le app in esecuzione su istanze EC2
- Gestione dell'accesso SSH
- Gestione degli aggiornamenti della sicurezza Linux
- Utilizzo dei gruppi di sicurezza

Gestione delle autorizzazioni utente di AWS OpsWorks Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Come best practice, limita gli utenti di AWS OpsWorks Stacks a un insieme specifico di azioni o a un insieme di risorse dello stack. Puoi controllare le autorizzazioni degli utenti di AWS OpsWorks Stacks in due modi: utilizzando la pagina Autorizzazioni AWS OpsWorks Stacks e applicando una policy IAM appropriata.

La pagina OpsWorks Autorizzazioni, o le azioni CLI o API equivalenti, consente di controllare le autorizzazioni utente in un ambiente multiutente in base allo stack assegnando a ciascun utente uno dei diversi livelli di autorizzazione. Ogni livello concede le autorizzazioni per un set standard di operazioni per una determinata risorsa dello stack. L'utilizzo della pagina Permissions (Autorizzazioni) consente di controllare le seguenti informazioni:

- Chi può accedere a ogni stack.
- Quali azioni ciascun utente può eseguire su ogni stack.

  Ad esempio, puoi consentire ad alcuni utenti solo di visualizzare lo stack, mentre altri possono distribuire applicazioni, aggiungere istanze e così via.
- Chi può gestire ogni stack.

  Puoi delegare la gestione di ogni stack a uno o più utenti specificati.

- Chi ha accesso SSH a livello utente e privilegi sudo (Linux) o accesso RDP e privilegi di amministratore (Windows) sulle istanze Amazon di ogni stack. EC2

  Puoi concedere o rimuovere queste autorizzazioni separatamente per ciascun utente in qualsiasi momento.

> **Important**
>
> La negazione dell'accesso SSH/RDP non necessariamente impedisce a un utente di eseguire l'accesso alle istanze. Se specifichi una coppia di EC2 chiavi Amazon per un'istanza, qualsiasi utente con la chiave privata corrispondente può accedere o utilizzare la chiave per recuperare la password dell'amministratore di Windows. Per ulteriori informazioni, consulta Gestione dell'accesso SSH.

Puoi utilizzare la console IAM, la CLI o l'API per aggiungere policy agli utenti che concedono autorizzazioni esplicite per le varie risorse e azioni AWS OpsWorks Stacks.

- L'utilizzo di una policy IAM per specificare le autorizzazioni è più flessibile rispetto all'utilizzo dei livelli di autorizzazione.
- Puoi configurare identità IAM (utenti, gruppi di utenti e ruoli), che concedono autorizzazioni alle identità IAM, come utenti e gruppi di utenti, o definire ruoli che possono essere associati a utenti federati.
- Una policy IAM è l'unico modo per concedere le autorizzazioni per determinate azioni chiave di Stacks. AWS OpsWorks

  Ad esempio, è necessario utilizzare IAM per concedere le autorizzazioni per `opsworks:CreateStack` e `opsworks:CloneStack`, che vengono utilizzate rispettivamente per creare e clonare gli stack.

Sebbene non sia esplicitamente possibile importare utenti federati nella console, un utente federato può creare implicitamente un profilo utente selezionando Le mie impostazioni nella parte superiore destra della console AWS OpsWorks Stacks e quindi scegliendo Utenti, sempre in alto a destra. Nella pagina Utenti, gli utenti federati, i cui account vengono creati utilizzando l'API o la CLI o implicitamente tramite la console, possono gestire i propri account in modo simile agli utenti non federati.

I due approcci non si escludono a vicenda e talvolta può risultare utile combinarli; AWS OpsWorks Stacks valuta quindi entrambi i set di autorizzazioni. Ad esempio, supponiamo di voler consentire agli utenti di aggiungere o eliminare istanze, ma non di aggiungere o eliminare livelli. Nessuno dei livelli di autorizzazione Stack concede quel set specifico di autorizzazioni. AWS OpsWorks Tuttavia, puoi utilizzare la pagina Autorizzazioni per concedere agli utenti un livello di autorizzazione di gestione, che consente loro di eseguire la maggior parte delle operazioni sullo stack, e quindi applicare una policy IAM che nega le autorizzazioni per aggiungere o rimuovere livelli. Per ulteriori informazioni, consulta Controllo dell'accesso alle AWS risorse mediante le policy.

Di seguito è riportato un modello tipico per la gestione delle autorizzazioni utente. In ogni caso, si presuppone che il lettore (tu) sia un utente amministratore.

1. Utilizza la console IAM per applicare AWSOpsWorks_FullAccess le policy a uno o più utenti amministrativi.
2. Crea un utente per ogni utente non amministrativo con una policy che non conceda autorizzazioni AWS OpsWorks Stacks.

   Se un utente richiede l'accesso solo a AWS OpsWorks Stacks, potrebbe non essere necessario applicare affatto una policy. Puoi invece gestire le loro autorizzazioni con la pagina AWS OpsWorks Stacks Permissions.
3. Utilizza la pagina AWS OpsWorks Stacks Users per importare gli utenti non amministrativi in Stacks. AWS OpsWorks
4. Per ogni stack, utilizzare la pagina Permissions (Autorizzazioni) per assegnare un livello di autorizzazione a ciascun utente.
5. Se necessario, personalizza i livelli di autorizzazione degli utenti applicando una policy IAM configurata in modo appropriato.

Per ulteriori consigli sulla gestione degli utenti, consulta. Best practice: Gestione delle autorizzazioni

Per ulteriori informazioni sulle best practice IAM, consulta la sezione Best practice di sicurezza in IAM nella IAM User Guide.

Argomenti

- Gestione degli AWS OpsWorks utenti di Stacks
- Concessione delle AWS OpsWorks autorizzazioni per stack agli utenti di Stacks
- Gestione delle autorizzazioni AWS OpsWorks di Stacks allegando una policy IAM

- Policy di esempio
- AWS OpsWorks Stacks (livelli di autorizzazione).

Gestione degli AWS OpsWorks utenti di Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Prima di poter importare gli utenti in AWS OpsWorks Stacks e concedere loro le autorizzazioni, devi prima aver creato un utente per ogni individuo. Per creare utenti IAM, inizia accedendo AWS come utente a cui sono state concesse le autorizzazioni definite nella IAMFull politica di accesso. Utilizza quindi la console IAM per creare utenti IAM per tutti coloro che devono accedere a AWS OpsWorks Stacks. Puoi quindi importare quegli utenti in AWS OpsWorks Stacks e concedere le autorizzazioni agli utenti come segue:

Utenti regolari di Stacks AWS OpsWorks

Gli utenti normali non richiedono una policy collegata. Se ne hanno uno, in genere non include alcuna autorizzazione AWS OpsWorks Stacks. Utilizza invece la pagina AWS OpsWorks Stacks Autorizzazioni per assegnare uno dei seguenti livelli di autorizzazione agli utenti normali su base regolare. stack-by-stack

- Le autorizzazioni Show (Visualizzazione) consentono agli utenti di visualizzare lo stack, ma non di eseguire operazioni.
- Le autorizzazioni Deploy (Distribuzione) includono le autorizzazioni Show (Visualizzazione) e consentono agli utenti di distribuire e aggiornare le app.
- Le autorizzazioni di gestione includono le autorizzazioni Deploy e consentono inoltre agli utenti di eseguire operazioni di gestione dello stack, come aggiungere livelli o istanze, utilizzare la pagina Autorizzazioni per impostare le autorizzazioni degli utenti e abilitare i propri privilegi. SSH/RDP and sudo/admin
- Le autorizzazioni Deny (Rifiuto) negano l'accesso allo stack.

Se questi livelli di autorizzazione non sono esattamente quelli che desideri per un determinato utente, puoi personalizzare le autorizzazioni dell'utente applicando una policy IAM. Ad esempio, potresti voler utilizzare la pagina AWS OpsWorks Pile di autorizzazioni per assegnare il livello di gestione delle autorizzazioni a un utente, che concede loro le autorizzazioni per eseguire tutte le operazioni di gestione degli stack, ma non per creare o clonare gli stack. È quindi possibile applicare una politica che limiti tali autorizzazioni negando loro l'autorizzazione ad aggiungere o eliminare livelli o aumenti tali autorizzazioni consentendo loro di creare o clonare pile. Per ulteriori informazioni, consulta Gestione delle autorizzazioni AWS OpsWorks di Stacks allegando una policy IAM.

AWS OpsWorks Stacks (utenti amministrativi)

Gli utenti amministrativi sono il proprietario dell'account o un utente IAM con le autorizzazioni definite dalla AWSOpsWorks_FullAccess policy. Oltre alle autorizzazioni concesse agli utenti con il livello Manage (Gestione), questa policy include le autorizzazioni relative alle operazioni che non possono essere concesse tramite la pagina Permissions (Autorizzazioni), ad esempio:

- Importazione di utenti in Stacks AWS OpsWorks
- Creazione e clonazione di stack

Per ulteriori informazioni sulle policy, consulta Policy di esempio. Per un elenco dettagliato delle autorizzazioni che possono essere concesse agli utenti solo applicando una policy IAM, consulta. AWS OpsWorks Stacks (livelli di autorizzazione).

Argomenti

- Utenti e regioni
- Creazione di un utente amministrativo di AWS OpsWorks Stacks
- Creazione di utenti IAM per Stacks AWS OpsWorks
- Importazione di utenti in pile AWS OpsWorks
- Modifica delle impostazioni utente di AWS OpsWorks Stacks

Utenti e regioni

AWS OpsWorks Gli utenti Stacks sono disponibili all'interno dell'endpoint regionale in cui sono stati creati. È possibile creare utenti in una delle seguenti regioni.

- Stati Uniti orientali (Ohio)

- Stati Uniti orientali (Virginia settentrionale)
- Regione Stati Uniti occidentali (Oregon)
- Regione Stati Uniti occidentali (California settentrionale)
- Solo regione (API) del Canada (Centrale); non disponibile nella AWS Management Console
- Regione Asia Pacifico (Mumbai)
- Regione Asia Pacifico (Singapore)
- Regione Asia Pacifico (Sydney)
- Regione Asia Pacifico (Tokyo)
- Regione Asia Pacifico (Seoul)
- Regione Europa (Francoforte)
- Regione Europa (Irlanda)
- Regione Europa (Londra)
- Regione Europa (Parigi)
- Regione Sud America (San Paolo)

Quando importi utenti in AWS OpsWorks Stacks, li importi in uno degli endpoint regionali; se desideri che un utente sia disponibile in più di una regione, devi importarlo in quella regione. Puoi anche importare utenti AWS OpsWorks Stacks da una regione all'altra; se importi un utente in una regione che ha già un utente con lo stesso nome, l'utente importato sostituisce l'utente esistente. Per ulteriori informazioni sull'importazione di utenti, consulta Importazione di utenti.

Creazione di un utente amministrativo di AWS OpsWorks Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi creare un utente amministrativo di AWS OpsWorks Stacks aggiungendo la `AWSOpsWorks_FullAccess` policy a un utente, che concede a quell'utente le autorizzazioni di

accesso completo a AWS OpsWorks Stacks. Per ulteriori informazioni sulla creazione di un utente amministrativo, consulta Creare un utente amministrativo.

> **Note**
>
> La AWSOpsWorks_FullAccess policy consente agli utenti di creare e gestire gli AWS OpsWorks stack Stacks, ma gli utenti non possono creare un ruolo di servizio IAM per lo stack; devono utilizzare un ruolo esistente. Il primo utente che crea uno stack deve disporre di autorizzazioni IAM aggiuntive, come descritto in. Autorizzazioni di amministrazione Quando questo utente crea il primo stack, AWS OpsWorks Stacks crea un ruolo di servizio IAM con le autorizzazioni richieste. Successivamente, qualsiasi utente con le autorizzazioni `opsworks:CreateStack` può utilizzare tale ruolo per creare altri stack. Per ulteriori informazioni, consulta Consentire a AWS OpsWorks Stacks di agire per tuo conto.

Quando crei un utente, puoi aggiungere ulteriori policy gestite dal cliente per ottimizzare le autorizzazioni dell'utente, se necessario. Ad esempio, potresti volere che un utente amministratore sia in grado di creare o eliminare gli stack, ma non di importare nuovi utenti. Per ulteriori informazioni, consulta Gestione delle autorizzazioni AWS OpsWorks di Stacks allegando una policy IAM.

Se hai più utenti amministrativi, invece di impostare le autorizzazioni separatamente per ogni utente, puoi aggiungere la AWSOpsWorks_FullAccess policy a un gruppo IAM e aggiungere gli utenti a quel gruppo.

Per informazioni sulla creazione di un gruppo, consulta Creazione di gruppi di utenti IAM. Quando crei il gruppo, aggiungi la AWSOpsWorks_FullAccesspolicy. Puoi anche aggiungere la AdministratorAccesspolitica, che include le AWSOpsWorks_FullAccessautorizzazioni.

Per informazioni sull'aggiunta di autorizzazioni a un gruppo esistente, consulta Allegare una policy a un gruppo di utenti IAM.

Creazione di utenti IAM per Stacks AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Prima di poter importare gli utenti IAM in AWS OpsWorks Stacks, devi crearli. Puoi farlo utilizzando la console IAM, la riga di comando o l'API. Per istruzioni complete, consulta Creazione di un utente IAM nel tuo AWS account.

A differenza degli utenti amministratori, non è necessario collegare una policy per definire le autorizzazioni. Puoi impostare le autorizzazioni dopo avere importato gli utenti in AWS OpsWorks Stacks, come descritto in Gestione delle autorizzazioni utente.

Per ulteriori informazioni sulla creazione di utenti e gruppi IAM, consulta Guida introduttiva a IAM.

Importazione di utenti in pile AWS OpsWorks

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

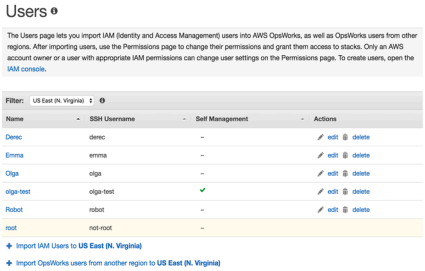
Gli utenti amministrativi possono importare utenti in AWS OpsWorks Stacks; possono anche importare utenti AWS OpsWorks Stacks da un endpoint regionale a un altro. Quando importi utenti in AWS OpsWorks Stacks, li importi in uno degli endpoint regionali di Stacks. AWS OpsWorks Se desideri che un utente sia disponibile in più di una regione, devi importarlo in quella regione.

Sebbene non sia esplicitamente possibile importare utenti federati nella console, un utente federato può creare implicitamente un profilo utente selezionando Le mie impostazioni nella parte superiore destra della console AWS OpsWorks Stacks e quindi selezionando Utenti, sempre in alto a destra. Nella pagina Utenti, gli utenti federati, i cui account vengono creati utilizzando l'API o la CLI o implicitamente tramite la console, possono gestire i propri account in modo simile agli utenti non federati.

Per AWS OpsWorks importare utenti in Stacks

1. Accedi a AWS OpsWorks Stacks come utente amministrativo o come proprietario dell'account.

2. Scegliere Users (Utenti) in alto a destra per aprire la pagina Users (Utenti).



3. Scegli Importa utenti IAM in < ***region name*** > per visualizzare gli utenti disponibili, ma che non sono ancora stati importati.



4. Selezionare la casella di controllo Select all (Seleziona tutto) oppure selezionare uno o più singoli utenti. Quando hai finito, scegli Importa in OpsWorks.

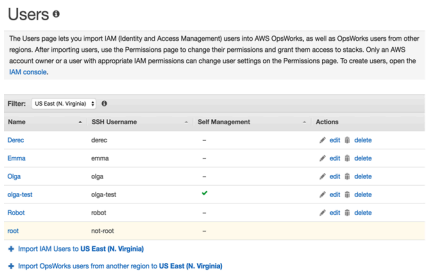
> **Note**
>
> Dopo aver importato un utente in AWS OpsWorks Stacks, se utilizzi la console o l'API IAM per eliminare l'utente dal tuo account, l'utente non perde automaticamente l'accesso SSH concesso tramite AWS OpsWorks Stacks. È inoltre necessario eliminare l'utente da AWS OpsWorks Stacks aprendo la pagina Utenti e selezionando delete nella colonna Azioni dell'utente.

Per importare gli utenti di AWS OpsWorks Stacks da una regione all'altra

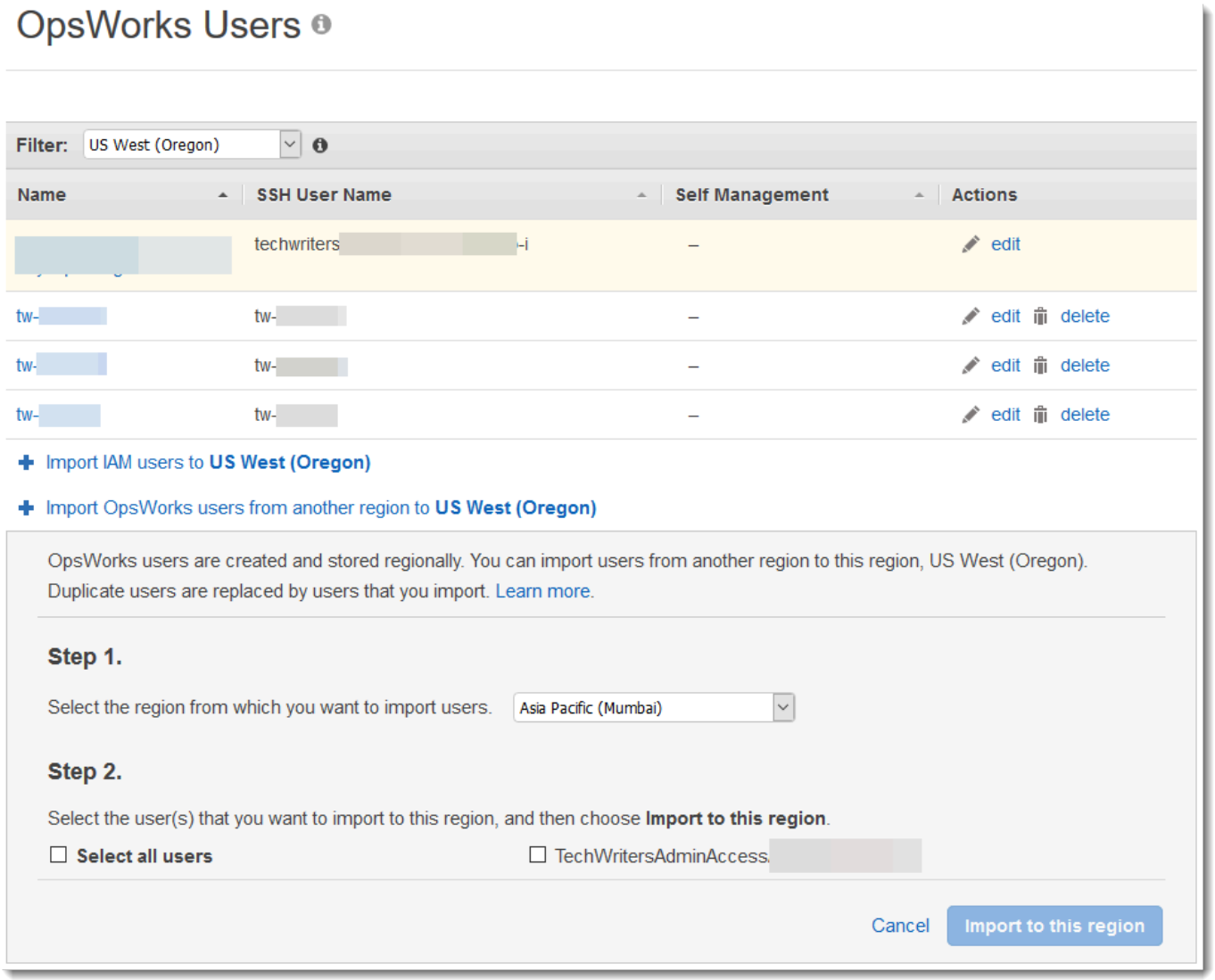
AWS OpsWorks Gli utenti Stacks sono disponibili all'interno dell'endpoint regionale in cui sono stati creati. È possibile creare utenti nelle regioni mostrate in. Utenti e regioni

Puoi importare gli utenti di AWS OpsWorks Stacks da una regione alla regione in cui è attualmente filtrato l'elenco degli utenti. Se importi un utente in una regione che ha già un utente con lo stesso nome, l'utente importato sostituisce l'utente esistente.

1. Accedi a AWS OpsWorks Stacks come utente amministrativo o come proprietario dell'account.
2. Scegliere Users (Utenti) in alto a destra per aprire la pagina Users (Utenti). Se hai utenti AWS OpsWorks Stacks in più di una regione, usa il controllo Filtro per filtrare in base alla regione in cui desideri importare gli utenti.



3. Scegli Importa utenti AWS OpsWorks Stacks da un'altra regione in < > ***current region***.



4. Seleziona la regione da cui desideri importare gli utenti di AWS OpsWorks Stacks.
5. Selezionare uno o più utenti da importare oppure selezionare tutti gli utenti e quindi scegliere Import to this region (Importa in questa regione). Attendi che AWS OpsWorks Stacks visualizzi gli utenti importati nell'elenco Utenti.

Unix IDs e utenti creati all'esterno degli stack AWS OpsWorks

AWS OpsWorks assegna agli utenti sulle istanze AWS OpsWorks Stacks valori Unix ID (UID) compresi tra 2000 e 4000. Perché AWS OpsWorks riserva l'intervallo 2000-4000 di UIDs, gli utenti che crei al di fuori di AWS OpsWorks (utilizzando ricette di libri di cucina o importando utenti da IAM, ad esempio) possono far sì che questi vengano AWS OpsWorks sovrascritti da Stacks per un UIDs altro utente. AWS OpsWorks Ciò può far sì che gli utenti che hai creato al di fuori di AWS OpsWorks Stacks non vengano visualizzati nei risultati di ricerca dei data bag o vengano esclusi dall'operazione integrata di Stacks. AWS OpsWorks `sync_remote_users`

I processi esterni possono anche creare utenti UIDs che AWS OpsWorks Stacks può sovrascrivere. Alcuni pacchetti del sistema operativo, ad esempio, possono creare un utente nell'ambito dei processi di post-installazione. Quando tu o un processo software create un utente su un sistema operativo basato su Linux senza specificare esplicitamente un UID, che è l'impostazione predefinita, l'UID assegnato da Stacks è + 1. AWS OpsWorks *`<highest existing AWS OpsWorks UID>`*

Come best practice, crea utenti Stacks e gestisci il loro accesso nella AWS OpsWorks console Stacks o utilizzando un SDK. AWS OpsWorks AWS CLI AWS Se crei utenti su istanze AWS OpsWorks Stacks esterne a AWS OpsWorks, usa *`UnixID`* valori superiori a 4000.

Modifica delle impostazioni utente di AWS OpsWorks Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver importato gli utenti, puoi modificare le impostazioni, nel seguente modo:

Per modificare le impostazioni utente

1. Nella pagina Users (Utenti), scegliere edit (modifica) nella colonna Actions (Azioni) dell'utente.
2. È possibile specificare le seguenti impostazioni.

Self Management (Gestione autonoma)

Seleziona Sì per consentire all'utente di utilizzare la MySettings pagina per specificare la propria chiave SSH personale.

> **Note**
>
> Puoi anche abilitare l'autogestione aggiungendo una policy IAM all'identità IAM che concede le autorizzazioni per le azioni and. DescribeMyUserProfileUpdateMyUserProfile

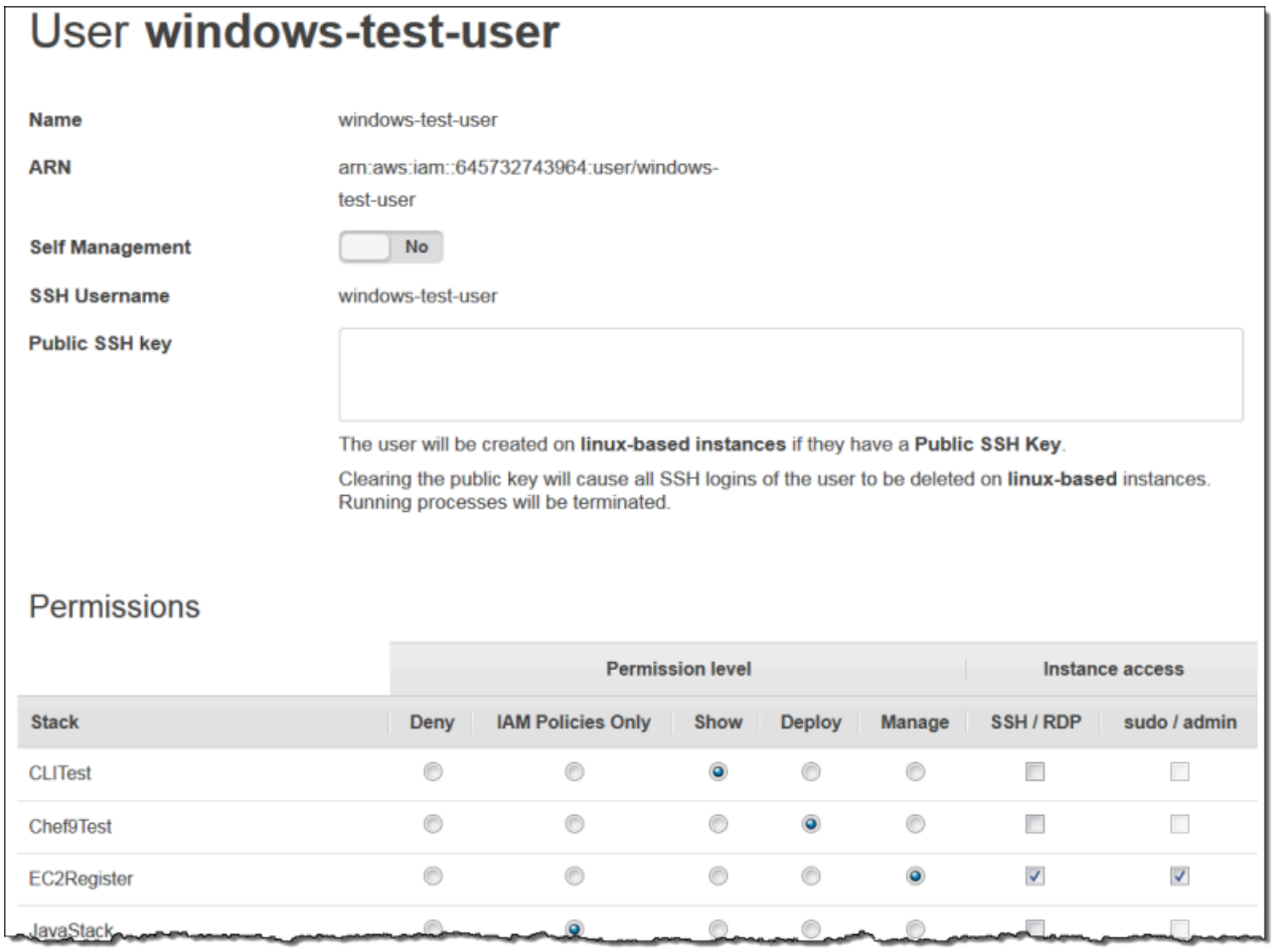
Chiave SSH pubblica

(Opzionale) Immettere una chiave SSH pubblica per l'utente. Questa chiave verrà visualizzata nella pagina My Settings (Le mie impostazioni) dell'utente. Se si abilita la gestione autonoma, l'utente può modificare la pagina My Settings (Le mie impostazioni) e specificare la propria chiave. Per ulteriori informazioni, consulta Registrazione della chiave SSH pubblica di un utente.

AWS OpsWorks Stacks installa questa chiave su tutte le istanze Linux; gli utenti possono utilizzare la chiave privata associata per accedere. Per ulteriori informazioni, consulta Accesso con SSH. Non è possibile utilizzare questa chiave con stack Windows.

Autorizzazioni

(Opzionale) Impostare i livelli di autorizzazione dell'utente per ogni stack in una posizione specifica anziché impostarli separatamente utilizzando la pagina Permissions (Autorizzazioni) di ogni stack. Per ulteriori informazioni sui livelli di autorizzazione, consulta Concessione delle autorizzazioni per stack.



Concessione delle AWS OpsWorks autorizzazioni per stack agli utenti di Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Il modo più semplice per gestire le autorizzazioni degli utenti di AWS OpsWorks Stack è utilizzare la pagina Autorizzazioni di uno stack. Ogni stack dispone della propria pagina, che consente di concedere autorizzazioni per lo stack specifico.

Devi effettuare l'accesso come utente amministratore o come utente con privilegi Manage (Gestione) per poter modificare le impostazioni delle autorizzazioni. L'elenco mostra solo gli utenti che sono stati importati in Stacks. AWS OpsWorks Per ulteriori informazioni su come creare e importare utenti, consulta Gestione degli utenti.

Il livello di autorizzazione predefinito è IAM Policies Only, che concede agli utenti solo le autorizzazioni incluse nella loro politica IAM.

- Quando importi un utente da IAM o da un'altra regione, l'utente viene aggiunto all'elenco per tutti gli stack esistenti con un livello di autorizzazione IAM Policies Only.
- Per impostazione predefinita, un utente che hai appena importato da un'altra regione non ha accesso agli stack nella regione di destinazione. Se importate utenti da un'altra regione, per consentire loro di gestire gli stack nella regione di destinazione, è necessario assegnare loro le autorizzazioni per tali stack dopo l'importazione degli utenti.
- Quando crei un nuovo stack, tutti gli utenti correnti vengono aggiunti all'elenco con i livelli di autorizzazione IAM Policies Only (Solo policy IAM).

Argomenti

- Impostazione delle autorizzazioni di un utente
- Visualizzazione delle autorizzazioni
- Utilizzo delle chiavi di condizione IAM per verificare le credenziali temporanee

Impostazione delle autorizzazioni di un utente

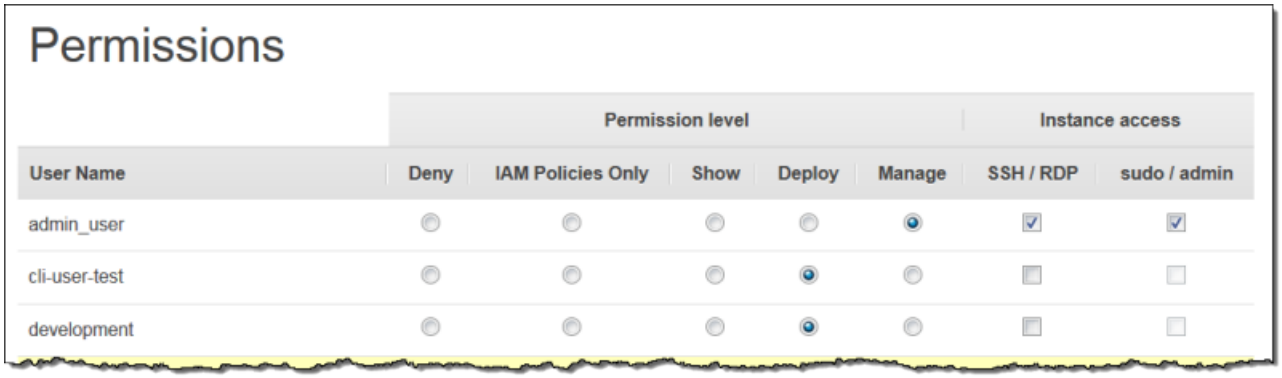
Per impostare le autorizzazioni di un utente

1. Nel riquadro di navigazione, seleziona Permissions (Autorizzazioni).
2. Nella pagina Permissions (Autorizzazioni), scegliere Edit (Modifica).
3. Modificare le impostazioni in Permission level (Livello di autorizzazione) Instance access (Accesso istanze):

    - Utilizzare le impostazioni di Permissions level (Livello di autorizzazione) per assegnare uno dei livelli di autorizzazione standard a ciascun utente. Ciò determina se l'utente può accedere allo stack corrente e quali azioni può eseguire. Se un utente dispone di una policy IAM, AWS OpsWorks Stacks valuta entrambi i set di autorizzazioni. Per un esempio, consulta Policy di esempio.

- In Instance access (Accesso istanze), l'impostazione SSH/RDP specifica se l'utente dispone dell'accesso SSH (Linux) o RDP (Windows) alle istanze dello stack.

  Se viene autorizzato l'accesso di tipo SSH/RDP, facoltativamente è possibile selezionare sudo/admin per concedere all'utente i privilegi sudo (Linux) o i privilegi di amministrazione (Windows) per le istanze dello stack.



A ciascun utente puoi assegnare uno dei seguenti livelli di autorizzazione. Per un elenco delle operazioni consentite da ogni livello, consulta AWS OpsWorks Stacks (livelli di autorizzazione)..

Rifiuta

L'utente non può eseguire alcuna azione AWS OpsWorks Stacks sullo stack, anche se dispone di una policy IAM che concede a AWS OpsWorks Stacks le autorizzazioni di accesso complete. Puoi utilizzare questo livello, ad esempio, per impedire ad alcuni utenti di accedere agli stack di prodotti non rilasciati.

IAM Policies Only (Solo policy IAM)

Si tratta del livello predefinito, che viene assegnato a tutti gli utenti appena importato e a tutti gli utenti per i nuovi stack creati. Le autorizzazioni dell'utente sono determinate dalla sua politica IAM. Se un utente non dispone di una policy IAM o la sua policy non dispone di autorizzazioni AWS OpsWorks Stacks esplicite, non può accedere allo stack. Agli utenti amministrativi viene in genere assegnato questo livello perché le loro politiche IAM concedono già autorizzazioni di accesso complete.

Mostra

L'utente può visualizzare uno stack, ma non eseguire operazioni. Ad esempio, i responsabili possono monitorare gli stack di un account, ma non hanno alcun bisogno di distribuire app o modificare lo stack.

Distribuzione

Include l'autorizzazione Show (Visualizzazione) e consente all'utente di distribuire applicazioni. Ad esempio, uno sviluppatore di app potrebbe aver bisogno di distribuire aggiornamenti alle istanze dello stack, ma non di aggiungere livelli o istanze allo stack.

Manage (Gestione)

Include le autorizzazioni Deploy (Distribuzione), ma consente all'utente anche di eseguire una serie di operazioni di gestione dello stack, tra cui:

- Aggiunta o eliminazione di livelli e istanze.
- Utilizzo della pagina Permissions (Autorizzazioni) per assegnare livelli di autorizzazione agli utenti.
- Registrazione o annullamento della registrazione delle risorse.

Ad esempio, ogni stack può essere associato a un responsabile designato, che deve verificare che lo stack includa il numero e il tipo appropriati di istanze, nonché gestire gli aggiornamenti di pacchetti e sistemi operativi e così via.

> **Note**
>
> Il livello Manage (Gestione) non consente agli utenti di creare o clonare gli stack. Tali autorizzazioni devono essere concesse da una policy IAM. Per vedere un esempio, consulta Gestione delle autorizzazioni.

Se l'utente dispone anche di una policy IAM, AWS OpsWorks Stacks valuta entrambi i set di autorizzazioni. Ciò consente di assegnare un livello di autorizzazione a un utente e quindi applicare una politica per limitare o aumentare le azioni consentite del livello. Ad esempio, è possibile applicare una politica che consenta a un utente di Manage di creare o clonare pile o neghi a tale utente la possibilità di registrare o annullare la registrazione delle risorse. Per alcuni esempi di tali policy, consulta Policy di esempio.

> **Note**
>
> Se la policy dell'utente consente operazioni aggiuntive, il risultato potrebbe sovrascrivere le impostazioni della pagina Permissions (Autorizzazioni). Ad esempio, se un utente dispone di una politica che consente l'CreateLayerazione ma utilizza la pagina Autorizzazioni per specificare le autorizzazioni Deploy, all'utente è comunque consentito creare livelli. L'eccezione a questa regola è l'opzione Deny, che nega l'accesso allo stack anche agli utenti con policy. AWSOpsWorks_FullAccess Per ulteriori informazioni, vedere Controllo dell'accesso alle AWS risorse mediante le politiche.

Visualizzazione delle autorizzazioni

Se la funzionalità di gestione autonoma è abilitata, gli utenti possono visualizzare un riepilogo dei propri livelli di autorizzazione per ogni stack scegliendo My Settings (Le mie impostazioni) in alto a destra. Gli utenti possono accedere alle mie impostazioni anche se la loro politica concede le autorizzazioni per le azioni DescribeMyUserProfilee UpdateMyUserProfile.

Utilizzo delle chiavi di condizione IAM per verificare le credenziali temporanee

AWS OpsWorks Stacks dispone di un livello di autorizzazione integrato che supporta casi di autorizzazione aggiuntivi (come la gestione semplificata dell'accesso in sola lettura o lettura-scrittura agli stack per i singoli utenti). Questo livello di autorizzazione dipende dall'utilizzo delle credenziali temporanee. Per questo motivo, non è possibile utilizzare una `aws:TokenIssueTime` condizione per verificare che gli utenti utilizzino credenziali a lungo termine o bloccare le azioni degli utenti che utilizzano credenziali temporanee, come descritto nel riferimento agli elementi delle policy di IAM JSON nella documentazione IAM.

Gestione delle autorizzazioni AWS OpsWorks di Stacks allegando una policy IAM

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi specificare le autorizzazioni AWS OpsWorks Stacks di un utente allegando una policy IAM. Per alcune autorizzazioni è necessario utilizzare una policy collegata:

- Autorizzazioni per gli utenti amministratore, ad esempio l'importazione di utenti.
- Autorizzazioni per alcune operazioni, ad esempio la creazione o la clonazione di uno stack.

Per un elenco completo delle operazioni che richiedono una policy collegata, consulta AWS OpsWorks Stacks (livelli di autorizzazione)..

Puoi anche utilizzare una policy per personalizzare i livelli di autorizzazione concessi tramite la pagina Autorizzazioni. Questa sezione fornisce un breve riepilogo di come applicare una policy IAM a un utente per specificare le autorizzazioni AWS OpsWorks Stacks. Per ulteriori informazioni, consulta Gestione degli accessi alle risorse. AWS

Una policy IAM è un oggetto JSON che contiene una o più istruzioni. Ogni istruzione include un elenco di autorizzazioni, che sono caratterizzate da tre elementi di base specifici:

Azione

Si tratta delle operazioni interessate dall'autorizzazione. Le azioni AWS OpsWorks Stacks vengono specificate come. `opsworks:`*`action`* L'elemento `Action` può essere impostato su un'operazione specifica, ad esempio `opsworks:CreateStack`, che specifica se l'utente può eseguire una chiamata a `CreateStack`. Puoi anche utilizzare i caratteri jolly per specificare gruppi di operazioni. Ad esempio, `opsworks:Create*` specifica tutte le operazioni di creazione. Per un elenco completo delle azioni AWS OpsWorks Stacks, consulta lo AWS OpsWorks Stacks API Reference.

Effetto

Indica se le operazioni specificate sono consentite o meno.

Risorsa

Le AWS risorse interessate dall'autorizzazione. AWS OpsWorks Stacks ha un tipo di risorsa, lo stack. Per specificare le autorizzazioni per una determinata risorsa stack, imposta `Resource` sull'ARN dello stack, che ha il formato seguente: `arn:aws:opsworks:`*`region`*`:`*`account_id`*`:stack/`*`stack_id`*`/`.

Puoi anche utilizzare i caratteri jolly. Ad esempio, imposta `Resource` su `*` per concedere le autorizzazioni per tutte le risorse.

Ad esempio, la seguente policy nega all'utente la possibilità di arrestare le istanze sullo stack il cui ID è `2860-2f18b4cb-4de5-4429-a149-ff7da9f0d8ee`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "opsworks:StopInstance",
      "Effect": "Deny",
      "Resource": "arn:aws:opsworks:*:*:stack/2f18b4cb-4de5-4429-a149-ff7da9f0d8ee/"
    }
  ]
}
```

Per informazioni sull'aggiunta di autorizzazioni a un utente IAM, consulta. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_change-permissions.html#users_change_permissions-add-console

Per ulteriori informazioni su come creare o modificare le politiche IAM, consulta Politiche e autorizzazioni in IAM. Per alcuni esempi di policy AWS OpsWorks Stacks, consulta. Policy di esempio

Policy di esempio

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione descrive esempi di policy IAM che possono essere applicate agli utenti di AWS OpsWorks Stacks.

- Autorizzazioni di amministrazionedescrive le politiche utilizzate per concedere le autorizzazioni agli utenti amministrativi.
- Gestione delle autorizzazionie Autorizzazioni Deploy (Distribuzione) mostra esempi di politiche che possono essere applicate a un utente per aumentare o limitare i livelli di autorizzazione Manage and Deploy.

AWS OpsWorks Stacks determina le autorizzazioni dell'utente valutando le autorizzazioni concesse dalle policy IAM e le autorizzazioni concesse dalla pagina Autorizzazioni. Per ulteriori informazioni, consulta Controllo dell'accesso alle risorse mediante le politiche. AWS Per ulteriori informazioni sulle autorizzazioni disponibili nella pagina Permissions (Autorizzazioni), consulta AWS OpsWorks Stacks (livelli di autorizzazione)..

Autorizzazioni di amministrazione

Utilizza la console IAM https://console.aws.amazon.com/iam/, per accedere alla AWSOpsWorks_FullAccess policy, collega questa policy a un utente per concedergli le autorizzazioni per eseguire tutte le azioni AWS OpsWorks Stacks. Le autorizzazioni IAM sono necessarie, tra le altre cose, per consentire a un utente amministrativo di importare utenti.

Devi creare ruoli IAM che consentano a AWS OpsWorks Stacks di agire per tuo conto per accedere ad altre AWS risorse, come le EC2 istanze Amazon. Di solito gestisci questa attività facendo creare il primo stack a un utente amministrativo e lasciando che AWS OpsWorks Stacks crei il ruolo per te. Puoi quindi utilizzare tale ruolo per tutti gli stack successivi. Per ulteriori informazioni, consulta Consentire a AWS OpsWorks Stacks di agire per tuo conto.

L'utente amministrativo che crea il primo stack deve disporre delle autorizzazioni per alcune azioni IAM che non sono incluse nella policy. AWSOpsWorks_FullAccess Aggiungi le seguenti autorizzazioni alla `Actions` sezione della policy. Per una corretta sintassi JSON, assicurati di aggiungere virgole tra le azioni e di rimuovere la virgola finale alla fine dell'elenco delle azioni.

```
"iam:PutRolePolicy",
"iam:AddRoleToInstanceProfile",
"iam:CreateInstanceProfile",
"iam:CreateRole"
```

Gestione delle autorizzazioni

Il livello di autorizzazione Manage (Gestione) consente a un utente di eseguire una serie di operazioni di gestione dello stack, ad esempio aggiungere o eliminare i livelli. Questo argomento descrive diverse politiche che è possibile utilizzare per gestire gli utenti per aumentare o limitare le autorizzazioni standard.

Impedire a un utente con l'autorizzazione Manage (Gestione) di aggiungere o eliminare i livelli

Puoi limitare il livello di autorizzazioni di Gestione per consentire a un utente di eseguire tutte le azioni di Manage tranne l'aggiunta o l'eliminazione di livelli utilizzando la seguente policy IAM. Sostituisci *region account_id* e *stack_id* con valori appropriati alla tua configurazione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:CreateLayer",
        "opsworks:DeleteLayer"
      ],
      "Resource": "arn:aws:opsworks:region:account_id:stack/stack_id/"
    }
  ]
}
```

Consentire a un utente con l'autorizzazione Manage (Gestione) di creare o clonare gli stack

Il livello di gestione delle autorizzazioni non consente agli utenti di creare o clonare pile. Puoi modificare le autorizzazioni di gestione per consentire a un utente di creare o clonare stack applicando la seguente politica IAM. Sostituisci *region* e *account_id* con valori appropriati alla tua configurazione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRolePolicy",
        "iam:ListRoles",
        "iam:ListInstanceProfiles",
        "iam:ListUsers",
        "opsworks:DescribeUserProfiles",
        "opsworks:CreateUserProfile",
        "opsworks:DeleteUserProfile"
      ],
      "Resource": "*"
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:opsworks::account_id:stack/*/",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "opsworks.amazonaws.com"
        }
      }
    }
  ]
}
```

Impedire a un utente con l'autorizzazione Manage (Gestione) di registrare o annullare la registrazione delle risorse

Il livello Manage permissions consente all'utente di registrare e annullare la registrazione delle risorse di indirizzi IP di Amazon EBS ed Elastic con lo stack. Puoi limitare le autorizzazioni di gestione per consentire all'utente di eseguire tutte le azioni di gestione tranne la registrazione delle risorse applicando la seguente politica.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:RegisterVolume",
        "opsworks:RegisterElasticIp"
      ],
      "Resource": "*"
    }
  ]
}
```

Consentire a un utente con l'autorizzazione Manage (Gestione) di importare utenti

Il livello di gestione delle autorizzazioni non consente agli utenti di importare utenti in Stacks. AWS OpsWorks Puoi aumentare le autorizzazioni di gestione per consentire a un utente di importare ed

eliminare utenti applicando la seguente politica IAM. Sostituisci *region* e *account_id* con valori appropriati alla tua configurazione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRolePolicy",
        "iam:ListRoles",
        "iam:ListInstanceProfiles",
        "iam:ListUsers",
        "iam:PassRole",
        "opsworks:DescribeUserProfiles",
        "opsworks:CreateUserProfile",
        "opsworks:DeleteUserProfile"
      ],
      "Resource": "arn:aws:iam:region:account_id:user/*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "opsworks.amazonaws.com"
        }
      }
    }
  ]
}
```

Autorizzazioni Deploy (Distribuzione)

Il livello di autorizzazione Deploy (Distribuzione) non consente agli utenti di creare o eliminare le app. Puoi aumentare le autorizzazioni Deploy per consentire a un utente di creare ed eliminare app applicando la seguente policy IAM. Sostituisci *region* e *stack_id* con valori appropriati alla tua configurazione. *account_id*

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

```
        "opsworks:CreateApp",
        "opsworks:DeleteApp"
      ],
      "Resource": "arn:aws:opsworks:region:account_id:stack/stack_id/"
    }
  ]
}
```

AWS OpsWorks Stacks (livelli di autorizzazione).

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione elenca le azioni consentite dai livelli di autorizzazione Mostra, Implementa e Gestisci nella AWS OpsWorks pagina Pile di autorizzazioni. Include anche un elenco di azioni per le quali è possibile concedere autorizzazioni solo applicando una policy IAM all'utente.

Mostra

Il livello Show (Visualizzazione) consente l'uso dei comandi `Describe`*`XYZ`* con le seguenti eccezioni:

```
DescribePermissions
DescribeUserProfiles
DescribeMyUserProfile
DescribeStackProvisioningParameters
```

Se un utente amministratore ha abilitato la gestione autonoma per l'utente, gli utenti con l'autorizzazione Show (Visualizzazione) possono usare anche `DescribeMyUserProfile` e `UpdateMyUserProfile`. Per ulteriori informazioni sulla gestione autonoma, consulta Modifica delle impostazioni utente.

Distribuzione

Le seguenti operazioni sono consentite dal livello Deploy (Distribuzione), in aggiunta alle operazioni consentite dal livello Show (Visualizzazione).

```
CreateDeployment
UpdateApp
```

Manage (Gestione)

Le seguenti operazioni sono consentite dal livello Manage (Gestione), in aggiunta alle operazioni consentite dai livelli Deploy (Distribuzione) e Show (Visualizzazione).

```
AssignInstance
AssignVolume
AssociateElasticIp
AttachElasticLoadBalancer
CreateApp
CreateInstance
CreateLayer
DeleteApp
DeleteInstance
DeleteLayer
DeleteStack
DeregisterElasticIp
DeregisterInstance
DeregisterRdsDbInstance
DeregisterVolume
DescribePermissions
DetachElasticLoadBalancer
DisassociateElasticIp
GrantAccess
GetHostnameSuggestion
RebootInstance
RegisterElasticIp
RegisterInstance
RegisterRdsDbInstance
RegisterVolume
SetLoadBasedAutoScaling
SetPermission
SetTimeBasedAutoScaling
StartInstance
StartStack
```

```
StopInstance
StopStack
UnassignVolume
UpdateElasticIp
UpdateInstance
UpdateLayer
UpdateRdsDbInstance
UpdateStack
UpdateVolume
```

Autorizzazioni che richiedono una policy IAM

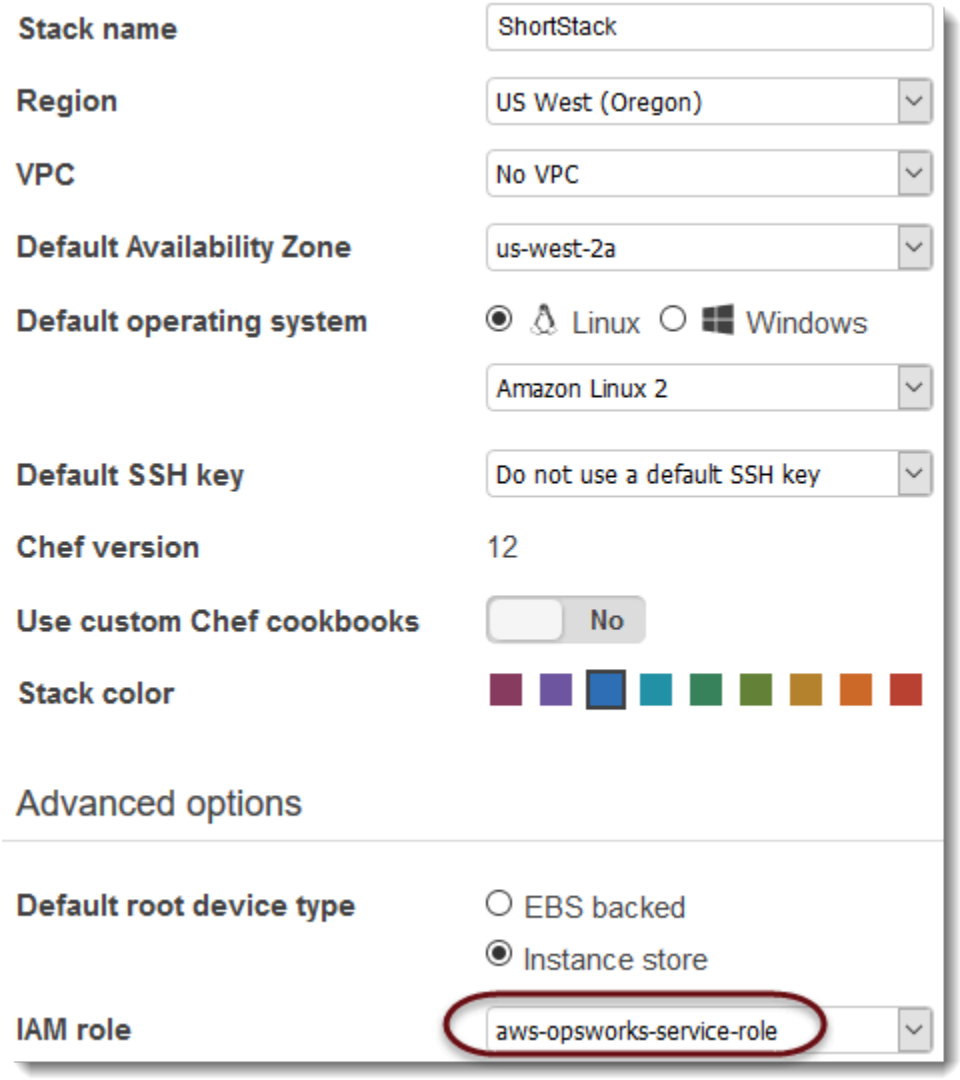
È necessario concedere le autorizzazioni per le seguenti azioni applicando all'utente una policy IAM appropriata. Per alcuni esempi, consulta Policy di esempio.

```
CloneStack
CreateStack
CreateUserProfile
DeleteUserProfile
DescribeUserProfiles
UpdateUserProfile
```

Consentire a AWS OpsWorks Stacks di agire per tuo conto

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks deve interagire con una varietà di servizi AWS per tuo conto. Ad esempio, AWS OpsWorks Stacks interagisce con Amazon EC2 per creare istanze e con Amazon CloudWatch per ottenere statistiche di monitoraggio. Quando crei uno stack, specifichi un ruolo IAM, di solito chiamato ruolo di servizio, che concede AWS OpsWorks a Stacks le autorizzazioni appropriate.



Quando specifichi il ruolo di servizio per un nuovo stack, puoi procedere in uno dei seguenti modi:

- Specificare un ruolo di servizio standard creato in precedenza.

  In genere, puoi creare un ruolo di servizio standard quando crei il tuo primo stack e quindi utilizzare tale ruolo per tutti gli stack successivi.
- Specificate un ruolo di servizio personalizzato creato utilizzando la console o l'API IAM.

  Questo approccio è utile se desideri concedere a AWS OpsWorks Stacks autorizzazioni più limitate rispetto al ruolo di servizio standard.

> **Note**
>
> Per creare il tuo primo stack, devi disporre delle autorizzazioni definite nel modello di policy IAM. AdministratorAccess Tali autorizzazioni consentono a AWS OpsWorks Stacks di creare un nuovo ruolo di servizio IAM e ti permettono di importare utenti, come descritto in precedenza. Per tutti gli stack successivi, gli utenti possono selezionare il ruolo di servizio creato per il primo stack. Non sono necessarie autorizzazioni amministrative complete per creare uno stack.

Il ruolo di servizio standard assegna le autorizzazioni seguenti:

- Esegui tutte EC2 le azioni Amazon (`ec2:*`).
- Ottieni CloudWatch statistiche (`cloudwatch:GetMetricStatistics`).
- Usa Elastic Load Balancing per distribuire il traffico ai server ()`elasticloadbalancing:*`.
- Usa un'istanza Amazon RDS come server di database (`rds:*`).
- Usa IAM roles (`iam:PassRole`) per fornire comunicazioni sicure tra AWS OpsWorks Stacks e le tue EC2 istanze Amazon.

Se crei un ruolo di servizio personalizzato, devi assicurarti che conceda tutte le autorizzazioni necessarie a AWS OpsWorks Stacks per gestire il tuo stack. Il seguente esempio JSON è l'istruzione della policy per il ruolo di servizio standard. Un ruolo di servizio personalizzato deve includere almeno le seguenti autorizzazioni nell'istruzione della policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:*",
                "iam:PassRole",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:DescribeAlarms",
                "ecs:*",
                "elasticloadbalancing:*",
                "rds:*"
            ],
            "Effect": "Allow",
```

```
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                }
            }
        }
    ]
}
```

Un ruolo di servizio ha inoltre una relazione di trust. I ruoli di servizio creati da AWS OpsWorks Stacks hanno la seguente relazione di fiducia.

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

Il ruolo di servizio deve avere questa relazione di fiducia affinché AWS OpsWorks Stacks agisca per tuo conto. Se utilizzi il ruolo di servizio predefinito, non modificare la relazione di trust. Se stai creando un ruolo di servizio personalizzato, specifica la relazione di fiducia effettuando una delle seguenti operazioni:

- Se stai utilizzando la procedura guidata Create role nella console IAM, in Scegli un caso d'uso, scegli Opsworks. Questo ruolo ha la relazione di fiducia appropriata, ma nessuna policy è associata implicitamente. Per concedere a AWS OpsWorks Stacks le autorizzazioni ad agire per tuo conto, crea una policy gestita dal cliente che contenga quanto segue e allegala al nuovo ruolo.

  ```
  {
  ```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricStatistics",
        "ec2:*",
        "ecs:*",
        "elasticloadbalancing:*",
        "iam:GetRolePolicy",
        "iam:ListInstanceProfiles",
        "iam:ListRoles",
        "iam:ListUsers",
        "rds:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "ec2.amazonaws.com"
        }
      }
    }
  ]
}
```

- Se utilizzi un AWS CloudFormation modello, puoi aggiungere qualcosa di simile a quanto segue alla sezione Risorse del modello.

```
"Resources": {
  "OpsWorksServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
```

```
            "Statement": [ {
              "Effect": "Allow",
              "Principal": {
                  "Service": [ "opsworks.amazonaws.com" ]
              },
              "Action": [ "sts:AssumeRole" ]
            } ]
        },
        "Path": "/",
        "Policies": [ {
            "PolicyName": "opsworks-service",
            "PolicyDocument": {
              ...
            } ]
        }
      },
   }
}
```

Prevenzione intersettoriale degli agenti confusi in Stacks AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Inoltre AWS, l'impersonificazione tra servizi diversi può portare al confuso problema del vicesceriffo. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a 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 ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account.

Ti consigliamo di utilizzare le chiavi di contesto `aws:SourceArn` e di condizione `aws:SourceAccount` globale nelle politiche di accesso allo stack per limitare le autorizzazioni che AWS OpsWorks Stacks fornisce agli stack con un altro servizio. Se il valore `aws:SourceArn` non contiene l'ID account, ad esempio un ARN di un bucket Amazon S3, è necessario utilizzare entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni. 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. Utilizza `aws:SourceArn` se desideri che venga associato un solo stack all'accesso tra servizi. `aws:SourceAccount`Utilizzalo se desideri consentire l'associazione di qualsiasi stack di quell'account all'utilizzo tra servizi.

Il valore di `aws:SourceArn` deve essere l'ARN di uno AWS OpsWorks stack.

Il modo più efficace per proteggersi dal confuso problema del vice consiste nell'utilizzare la chiave `aws:SourceArn` global condition context con l'ARN completo dello AWS OpsWorks stack Stacks. Se non conosci l'ARN completo o se stai specificando più stack ARNs, usa la chiave `aws:SourceArn` global context condition con wildcards (*) per le parti sconosciute dell'ARN. Ad esempio `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

La sezione seguente mostra come utilizzare le chiavi di contesto `aws:SourceArn` e `aws:SourceAccount` global condition in AWS OpsWorks Stacks per evitare il confuso problema del vice.

Evita gli exploit confusi dei vice in Stacks AWS OpsWorks

Questa sezione descrive come contribuire a prevenire la confusione degli exploit dei deputati in AWS OpsWorks Stacks e include esempi di policy di autorizzazione che puoi associare al ruolo IAM che stai utilizzando per accedere a Stacks. AWS OpsWorks Come best practice in materia di sicurezza, ti consigliamo di aggiungere le chiavi `aws:SourceArn` e `aws:SourceAccount` condition alle relazioni di fiducia che il tuo ruolo IAM intrattiene con altri servizi. Le relazioni di fiducia consentono a AWS OpsWorks Stacks di assumere il ruolo di eseguire azioni in altri servizi necessari per creare o gestire i tuoi AWS OpsWorks stack Stacks.

Per modificare le relazioni di fiducia per aggiungere **`aws:SourceArn`** e condizionare le chiavi **`aws:SourceAccount`**

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, seleziona Ruoli.

3. Nella casella di ricerca, cerca il ruolo che utilizzi per accedere a AWS OpsWorks Stacks. Il ruolo AWS gestito è`aws-opsworks-service-role`.
4. Nella pagina di riepilogo del ruolo, scegli la scheda Relazioni di fiducia.
5. Nella scheda Relazioni di fiducia, scegli Modifica politica di fiducia.
6. Nella pagina Modifica politica di fiducia, aggiungi almeno una delle chiavi `aws:SourceArn` o delle `aws:SourceAccount` condizioni alla politica. Utilizzalo `aws:SourceArn` per limitare la relazione di fiducia tra servizi incrociati (come Amazon EC2) e AWS OpsWorks Stacks a stack Stacks specifici AWS OpsWorks , il che è più restrittivo. Aggiungi `aws:SourceAccount` per limitare la relazione di fiducia tra cross services e AWS OpsWorks Stacks agli stack di un account specifico, il che è meno restrittivo. Di seguito è riportato un esempio. Tieni presente che se utilizzi entrambe le chiavi di condizione, l'account IDs deve essere lo stesso.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "opsworks.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnEquals": {
          "arn:aws:opsworks:us-east-2:123456789012:stack/
EXAMPLEd-5699-40a3-80c3-22c32EXAMPLE/"
        }
      }
    }
  ]
}
```

7. Quando hai finito di aggiungere le chiavi di condizione, scegli Aggiorna politica.

Di seguito sono riportati altri esempi di ruoli che limitano l'accesso agli stack utilizzando `aws:SourceArn` and`aws:SourceAccount`.

Argomenti

- Esempio: accesso agli stack in una regione specifica
- Esempio: aggiunta di più di uno stack ARN a aws:SourceArn

Esempio: accesso agli stack in una regione specifica

La seguente dichiarazione sulla relazione di fiducia tra i ruoli accede a qualsiasi AWS OpsWorks stack Stacks nella regione Stati Uniti orientali (Ohio) (). `us-east-2` Nota che la regione è specificata nel valore ARN di`aws:SourceArn`, ma il valore dell'ID dello stack è un carattere jolly (*).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "opsworks.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:opsworks:us-east-2:123456789012:stack/*"
        }
      }
    }
  ]
}
```

Esempio: aggiunta di più di uno stack ARN a **`aws:SourceArn`**

L'esempio seguente limita l'accesso a una matrice di due AWS OpsWorks stack di stack nell'account ID 123456789012.

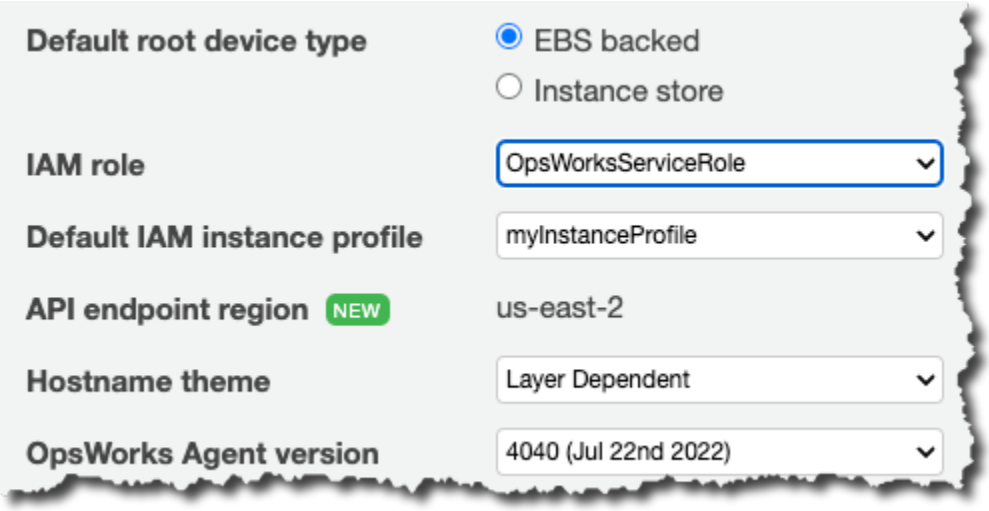
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "opsworks.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnEquals": {
          "aws:SourceArn": [
             "arn:aws:opsworks:us-east-2:123456789012:stack/unique_ID1",
             "arn:aws:opsworks:us-east-2:123456789012:stack/unique_ID2"
           ]
       }
      }
    }
  ]
}
```

Specificazione delle autorizzazioni per le app in esecuzione su istanze EC2

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se le applicazioni in esecuzione sulle EC2 istanze Amazon del tuo stack devono accedere ad altre risorse AWS, come i bucket Amazon S3, devono disporre delle autorizzazioni appropriate. Per conferire tali autorizzazioni, puoi utilizzare un profilo dell'istanza. Puoi specificare un profilo di istanza per ogni istanza quando crei uno stack Stacks. AWS OpsWorks



Puoi anche specificare un profilo per le istanze di un livello modificando la configurazione del livello.

Il profilo dell'istanza specifica un ruolo IAM. Le applicazioni in esecuzione sull'istanza possono assumere quel ruolo per accedere alle risorse AWS, in funzione permessi assegnati dalla policy di ruolo. Per ulteriori informazioni sull'assunzione di un ruolo da parte di un'applicazione, consulta la pagina sull'assunzione del ruolo tramite chiamata API.

È possibile creare un profilo dell'istanza in uno dei seguenti modi:

- Utilizza la console o l'API IAM per creare un profilo.

  Per ulteriori informazioni, consulta l'argomento relativo ai ruoli (delegazione e federazione).
- Usa un AWS CloudFormation modello per creare un profilo.

  Per alcuni esempi di come includere risorse IAM in un modello, consulta Identity and Access Management (IAM) Template Snippets.

Un profilo dell'istanza deve avere una relazione di trust e una policy collegata che assegni i permessi di accesso alle risorse AWS.

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

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

Il profilo dell'istanza deve avere questa relazione di fiducia affinché AWS OpsWorks Stacks agisca per tuo conto. Se utilizzi il ruolo di servizio predefinito, non modificare la relazione di trust. Se crei un ruolo di servizio personalizzato, specifica la relazione di trust come segue:

- Se utilizzi la procedura guidata Create Role nella console IAM, specifica il tipo di EC2 ruolo Amazon in AWS Service Roles nella seconda pagina della procedura guidata.
- Se utilizzi un AWS CloudFormation modello, puoi aggiungere qualcosa come segue alla sezione Risorse del modello.

```
"Resources": {
      "OpsWorksEC2Role": {
         "Type": "AWS::IAM::Role",
         "Properties": {
            "AssumeRolePolicyDocument": {
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "ec2.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            },
            "Path": "/"
         }
      },
      "RootInstanceProfile": {
         "Type": "AWS::IAM::InstanceProfile",
         "Properties": {
            "Path": "/",
            "Roles": [ {
               "Ref": "OpsWorksEC2Role"
            }
         ]
      }
   }
```

```
}
```

Quando crei il profilo dell'istanza, puoi allegare una policy appropriata al ruolo del profilo in quel momento. Dopo aver creato lo stack, devi utilizzare la console o l'API IAM per allegare una policy appropriata al ruolo del profilo. Ad esempio, la seguente policy garantisce l'accesso completo a tutti gli oggetti nel bucket Amazon S3 denominato amzn-s3-demo-bucket. *region*Sostituisci amzn-s3-demo-bucket con valori appropriati alla tua configurazione.

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

Per un esempio di come creare e utilizzare un profilo dell'istanza, consulta Utilizzo di un bucket Amazon S3.

Se l'applicazione utilizza un profilo di istanza per chiamare l'API AWS OpsWorks Stacks da un' EC2 istanza, la policy deve consentire l'`iam:PassRole`azione oltre alle azioni appropriate per AWS OpsWorks Stacks e altri servizi AWS. L'autorizzazione `iam:PassRole` consente a AWS OpsWorks Stacks di assumere il ruolo di servizio per tuo conto. Per ulteriori informazioni sull'API AWS OpsWorks Stacks, consulta AWS OpsWorks API Reference.

Di seguito è riportato un esempio di policy IAM che consente di richiamare qualsiasi azione AWS OpsWorks Stacks da un' EC2 istanza, nonché qualsiasi azione Amazon EC2 o Amazon S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*",
        "s3:*",
        "opsworks:*",
        "iam:PassRole"
```

```
      ],
      "Resource": "arn:aws:ec2:region:account_id:instance/*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "opsworks.amazonaws.com"
        }
      }
    }
  ]
}
```

> **Note**
>
> Se non lo consenti`iam:PassRole`, qualsiasi tentativo di richiamare un'azione AWS OpsWorks Stacks fallisce e genera un errore come il seguente:
>
> ```
> User: arn:aws:sts::123456789012:federated-user/Bob is not authorized
> to perform: iam:PassRole on resource:
> arn:aws:sts::123456789012:role/OpsWorksStackIamRole
> ```

Per ulteriori informazioni sull'utilizzo dei ruoli su un' EC2 istanza per le autorizzazioni, consulta Concedere alle applicazioni eseguite su Amazon EC2 Instances l'accesso alle risorse AWS nella Guida per l'AWS Identity and Access Management utente.

Gestione dell'accesso SSH

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks supporta le chiavi SSH per gli stack Linux e Windows.

- Per le istanze Linux, puoi utilizzare SSH per accedere a un'istanza, ad esempio, per l'esecuzione dei comandi della CLI dell'agente.

  Per ulteriori informazioni, consulta Accesso con SSH.
- Per le istanze Windows, puoi utilizzare una chiave SSH per ottenere la password dell'amministratore dell'istanza, che puoi quindi utilizzare per eseguire l'accesso con RDP.

  Per ulteriori informazioni, consulta Accesso con RDP.

L'autenticazione è basata su una coppia di chiavi SSH, composta da una chiave pubblica e una chiave privata:

- Installi la chiave pubblica sull'istanza.

  La posizione dipende dal particolare sistema operativo, ma AWS OpsWorks Stacks gestisce i dettagli al posto tuo.
- Puoi archiviare la chiave privata in locale e fornirla a un client SSH, ad esempio `ssh.exe`, per accedere all'istanza.

  Il client SSH utilizza la chiave privata per connettersi all'istanza.

Per fornire l'accesso SSH agli utenti di uno stack, devi creare coppie di chiavi SSH, installare le chiavi pubbliche sulle istanze dello stack e gestire in modo sicuro le chiavi private.

Amazon EC2 offre un modo semplice per installare una chiave SSH pubblica su un'istanza. Puoi utilizzare la EC2 console o l'API Amazon per creare una o più coppie di chiavi per ogni regione AWS che intendi utilizzare. Amazon EC2 archivia le chiavi pubbliche su AWS e tu memorizzi le chiavi private localmente. Quando avvii un'istanza, specifichi una delle coppie di chiavi della regione e Amazon la installa EC2 automaticamente sull'istanza. Utilizzi quindi la chiave privata corrispondente per eseguire l'accesso all'istanza. Per ulteriori informazioni, consulta Amazon EC2 Key Pairs.

Con AWS OpsWorks Stacks, puoi specificare una delle coppie di EC2 chiavi Amazon della regione quando crei uno stack e, facoltativamente, sovrascriverla con una coppia di chiavi diversa quando crei ogni istanza. Quando AWS OpsWorks Stacks avvia l' EC2 istanza Amazon corrispondente, specifica la coppia di chiavi e Amazon EC2 installa la chiave pubblica sull'istanza. Puoi quindi utilizzare la chiave privata per accedere o recuperare una password di amministratore, proprio come

faresti con un' EC2 istanza Amazon standard. Per ulteriori informazioni, consulta Installazione di una EC2 chiave Amazon.

L'utilizzo di una coppia di EC2 chiavi Amazon è comodo, ma presenta due limitazioni significative:

- Una coppia di EC2 chiavi Amazon è legata a una particolare regione AWS.

  In caso di utilizzo di più regioni, devi gestire più coppie di chiavi.
- Puoi installare solo una coppia di EC2 chiavi Amazon su un'istanza.

  Se vuoi consentire a più utenti di effettuare l'accesso, tali utenti devono avere una copia della chiave privata, cosa sconsigliata per motivi di sicurezza.

Per gli stack Linux, AWS OpsWorks Stacks offre un modo più semplice e flessibile per gestire le coppie di chiavi SSH.

- Ogni utente registra una coppia personale di chiavi.

  Memorizzano la chiave privata localmente e registrano la chiave pubblica con AWS OpsWorks Stacks, come descritto in. Registrazione della chiave SSH pubblica di un utente
- Quando imposti le autorizzazioni per uno stack, specifichi gli utenti che devono avere un accesso SSH alle istanze dello stack.

  AWS OpsWorks Stacks crea automaticamente un utente di sistema sulle istanze dello stack per ogni utente autorizzato e installa la relativa chiave pubblica. L'utente può quindi utilizzare la chiave privata corrispondente per effettuare l'accesso, come descritto in Accesso con SSH.

L'utilizzo di chiavi SSH personali presenta i vantaggi seguenti.

- Non è necessario configurare manualmente le chiavi sulle istanze; AWS OpsWorks Stacks installa automaticamente le chiavi pubbliche appropriate su ogni istanza.
- AWS OpsWorks Stacks installa solo le chiavi pubbliche personali degli utenti autorizzati.

  Gli utenti non autorizzati non possono utilizzare la propria chiave privata personali per accedere alle istanze. Con le coppie di EC2 chiavi Amazon, qualsiasi utente con la chiave privata corrispondente può accedere, con o senza accesso SSH autorizzato.
- Se un utente non necessita più dell'accesso SSH, puoi utilizzare la pagina Permissions (Autorizzazioni) per revocare le autorizzazioni SSH/RDP dell'utente.

AWS OpsWorks Stacks disinstalla immediatamente la chiave pubblica dalle istanze dello stack.

- Puoi utilizzare la stessa chiave per qualsiasi regione AWS.

  Gli utenti devono gestire solo una chiave privata.

- Non è necessario condividere le chiavi private.

  Ogni utente dispone della propria chiave privata.

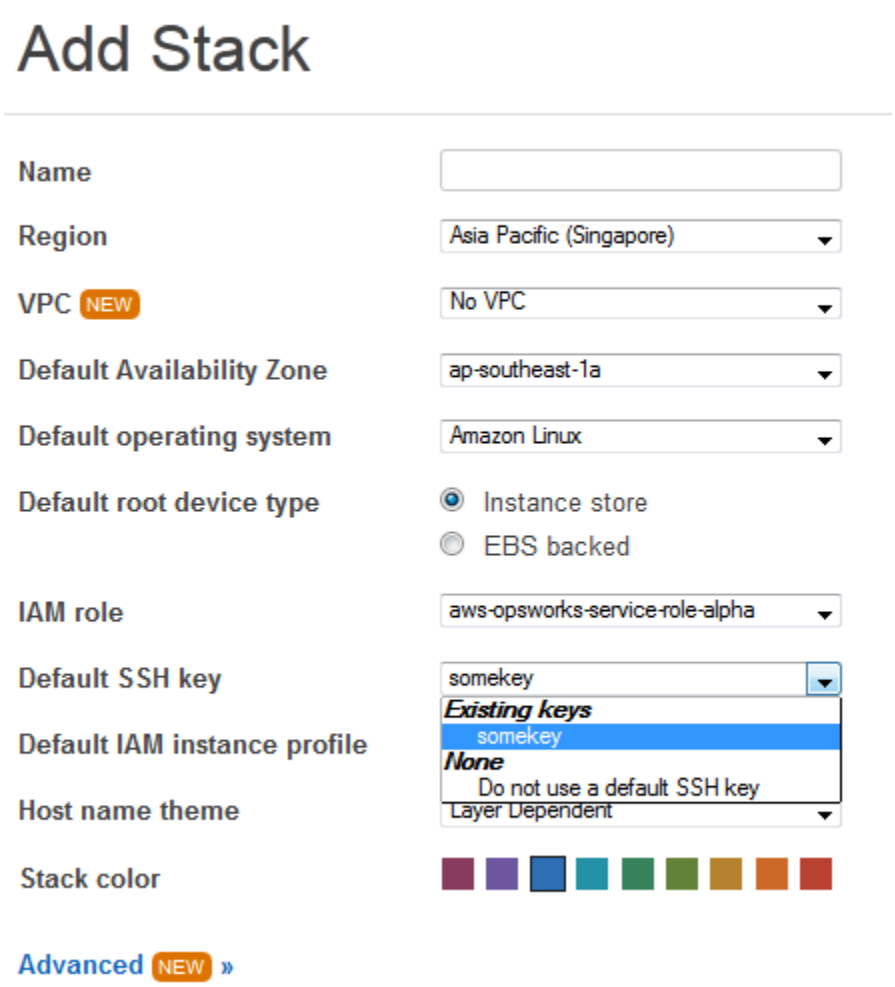
- È facile ruotare chiavi.

  La chiave pubblica viene aggiornata in My Settings (Le mie impostazioni) e AWS OpsWorks Stacks aggiorna automaticamente le istanze.

Installazione di una EC2 chiave Amazon

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

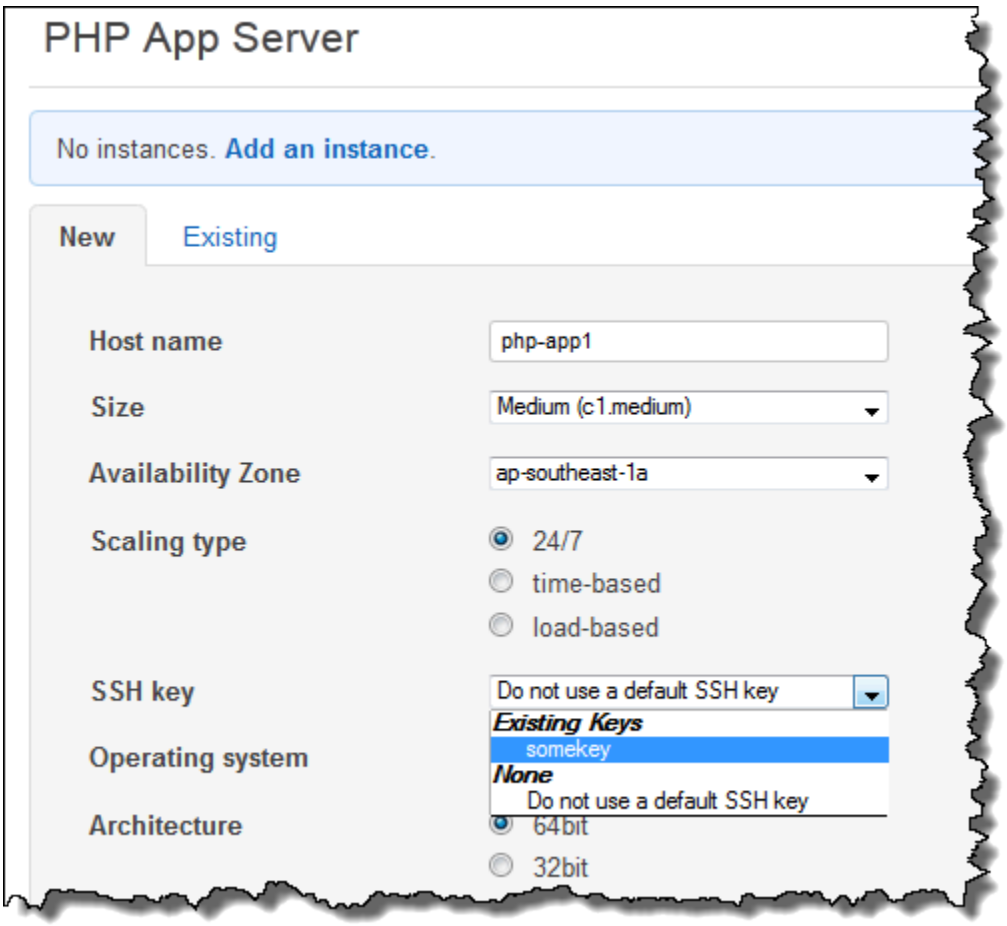
Quando crei uno stack, puoi specificare una chiave Amazon EC2 SSH che viene installata di default su ogni istanza dello stack.



L'elenco delle chiavi SSH predefinite mostra Amazon EC2keys del tuo account AWS. Puoi effettuare una delle seguenti operazioni:

- Selezionare la chiave appropriata dall'elenco.
- Selezionare Do not use a default SSH key (Non utilizzare una chiave SSH predefinita) per non specificare alcuna chiave.

Se hai selezionato Do not use a default SSH key (Non utilizzare una chiave SSH predefinita) oppure desideri sostituire la chiave predefinita di uno stack, puoi specificare una chiave quando crei un'istanza.



All'avvio dell'istanza, AWS OpsWorks Stacks installa la chiave pubblica nel file. `authorized_keys`

Registrazione della chiave SSH pubblica di un utente

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Sono disponibili due modi per registrare la chiave SSH pubblica di un utente:

- Un utente amministratore può assegnare una chiave SSH pubblica a uno o più utenti e fornire loro la chiave privata corrispondente.

- Un utente amministratore può abilitare la gestione autonoma per uno o più utenti.

  Gli utenti possono quindi specificare la propria chiave SSH pubblica.

Per ulteriori informazioni su come gli utenti amministratori possono abilitare la gestione autonoma o assegnare chiavi pubbliche agli utenti, consulta Modifica delle impostazioni utente.

La connessione a istanze basate su Linux mediante SSH in un terminale PuTTY richiede fasi aggiuntive. Per ulteriori informazioni, consulta Connessione all'istanza Linux da Windows tramite PuTTY e Risoluzione dei problemi di connessione all'istanza nella documentazione AWS.

Di seguito viene descritto come un utente con l'autogestione abilitata può specificare la propria chiave pubblica.

Per specificare la chiave SSH pubblica

1. Creare una coppia di chiavi SSH.

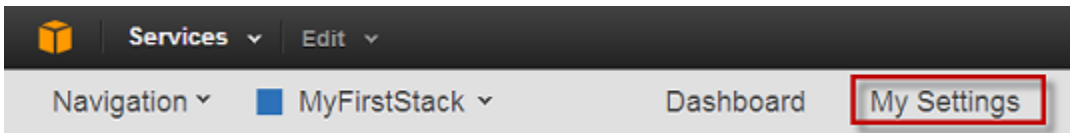
   L'approccio più semplice è generare la coppia di chiavi in locale. Per ulteriori informazioni, consulta Come generare la propria chiave e importarla su Amazon EC2.

   > **Note**
   >
   > Se usi Pu TTYgen per generare la tua coppia di chiavi, copia la chiave pubblica dalla chiave pubblica per incollarla nella casella del file OpenSSH authorized_keys. Facendo clic su Salva chiave pubblica si salva la chiave pubblica in un formato non supportato da. MindTerm

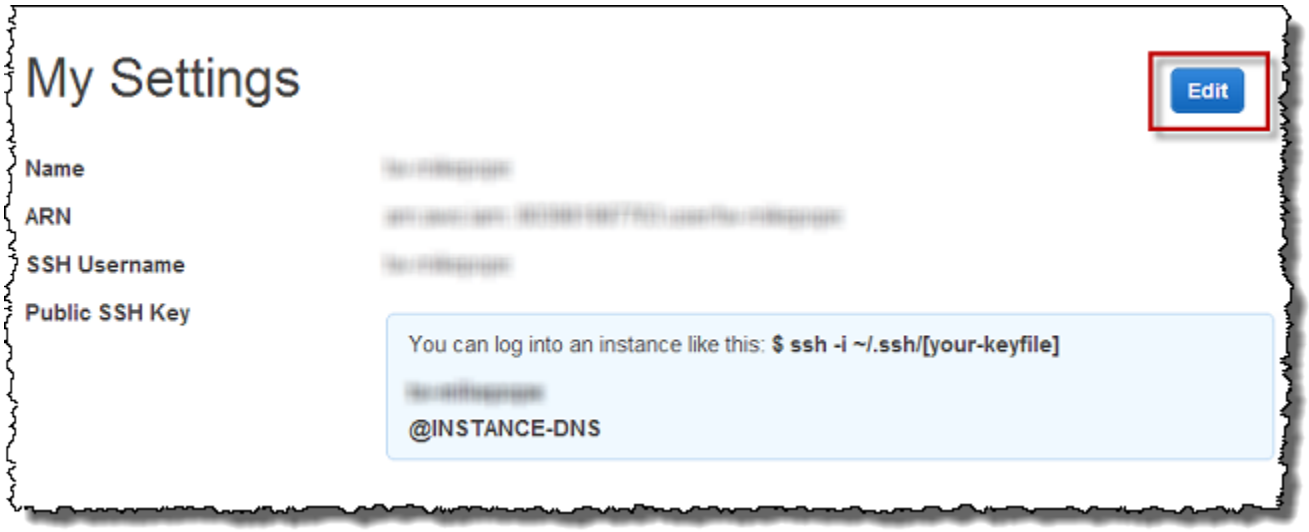
2. Accedi alla console AWS OpsWorks Stacks come utente IAM con l'autogestione abilitata.

   > **Important**
   >
   > Se accedi come proprietario dell'account o come utente IAM che non ha abilitato l'autogestione, AWS OpsWorks Stacks non visualizza Le mie impostazioni. Un utente amministratore o un proprietario dell'account può specificare le chiavi SSH. A tale scopo, è necessario passare alla pagina Users (Utenti) e modificare le impostazioni utente.

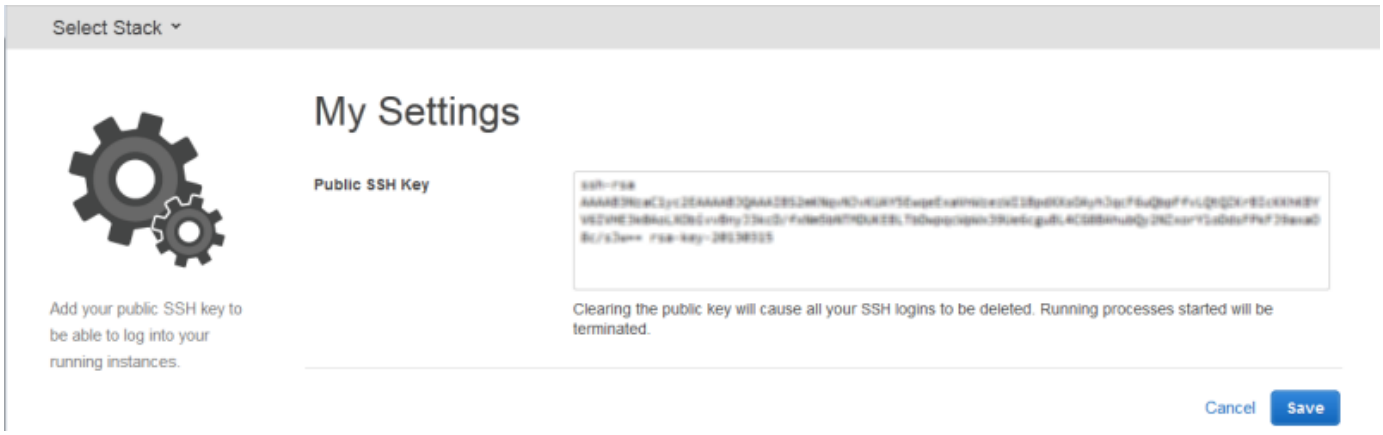
3. Seleziona Le mie impostazioni, che mostra le impostazioni per l'utente che ha effettuato l'accesso.



4. Nella pagina My Settings (Le mie impostazioni), fare clic su Edit (Modifica).



5. Nella casella Public SSH Key (Chiave SSH pubblica), immettere la chiave SSH pubblica, quindi fare clic su Save (Salva).



> **Important**
>
> Per utilizzare il client MindTerm SSH integrato per connettersi alle EC2 istanze Amazon, un utente deve aver effettuato l'accesso come utente IAM e disporre di una chiave SSH pubblica registrata con Stacks. AWS OpsWorks Per ulteriori informazioni, consulta Utilizzo del client SSH integrato MindTerm .

Gestione degli aggiornamenti della sicurezza Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Aggiornamenti di sicurezza

I provider del sistema operativo Linux rilasciano aggiornamenti regolari, la maggior parte dei quali sono patch di sicurezza del sistema operativo, ma che possono includere anche aggiornamenti dei pacchetti installati. Ti consigliamo di verificare che i sistemi operativi delle istanze siano aggiornati in base alle patch di sicurezza più recenti.

Per impostazione predefinita, AWS OpsWorks Stacks installa automaticamente gli ultimi aggiornamenti durante la configurazione, al termine dell'avvio di un'istanza. AWS OpsWorks Stacks non installa automaticamente gli aggiornamenti dopo che un'istanza è online, per evitare interruzioni come il riavvio dei server delle applicazioni. Al contrario, puoi gestire gli aggiornamenti delle istanze online in modo autonomo, in modo da ridurre al minimo i tempi di inattività.

Consigliamo di usare uno dei seguenti metodi per aggiornare le istanze online.

- Creare e avviare nuove istanze per sostituire le istanze online correnti. Eliminare quindi le istanze correnti.

  Nelle nuove istanze verrà installato il set più recente di patch di sicurezza durante la configurazione.
- Nelle istanze basate su Linux in Chef 11.10 o negli stack di versioni precedenti, eseguire il comando Update Dependencies (Aggiorna dipendenze) sullo stack per installare il set corrente di patch di sicurezza e altri aggiornamenti sulle istanze specificate.

Per entrambi questi approcci, AWS OpsWorks Stacks esegue l'aggiornamento eseguendo `yum update` per Amazon Linux e Red Hat Enterprise Linux (RHEL) o `apt-get update` per Ubuntu.

Ogni distribuzione gestisce gli aggiornamenti in modo diverso. Devi pertanto valutare le informazioni nei collegamenti associati per comprendere esattamente l'impatto di un aggiornamento sulle istanze:

- Amazon Linux: gli aggiornamenti di Amazon Linux installano patch di sicurezza e possono anche installare aggiornamenti delle funzionalità, inclusi gli aggiornamenti dei pacchetti.

  Per ulteriori informazioni, consulta AMI Amazon Linux FAQs.
- Ubuntu: gli aggiornamenti di Ubuntu si limitano in gran parte all'installazione di patch di sicurezza, ma possono anche installare aggiornamenti di pacchetti per un numero limitato di correzioni critiche.

  Per ulteriori informazioni, consulta LTS - Ubuntu Wiki.
- CentOS: gli aggiornamenti CentOS generalmente mantengono la compatibilità binaria con le versioni precedenti.
- RHEL — Gli aggiornamenti RHEL generalmente mantengono la compatibilità binaria con le versioni precedenti.

  Per ulteriori informazioni, consulta la pagina relativa al ciclo di vita Red Hat Enterprise Linux.

Se desideri un maggiore controllo sugli aggiornamenti, ad esempio specificando versioni particolari del pacchetto, puoi disabilitare gli aggiornamenti automatici utilizzando le UpdateLayerazioni CreateInstance, UpdateInstanceCreateLayer, o i metodi SDK AWS equivalenti o i comandi CLI AWS su cui impostare il parametro. `InstallUpdatesOnBoot` `false` L'esempio seguente mostra come usare AWS CLI per disabilitare `InstallUpdatesOnBoot` come impostazione predefinita per un livello esistente.

```
aws opsworks update-layer --layer-id layer ID --no-install-updates-on-boot
```

Devi quindi gestire gli aggiornamenti manualmente. Ad esempio, è impiegare una delle seguenti strategie:

- Implementare una ricetta personalizzata che esegua il comando shell appropriato per installare gli aggiornamenti desiderati.

  Poiché gli aggiornamenti di sistema non sono mappati naturalmente a un evento del ciclo di vita, includere la ricetta nei libri di ricette personalizzati, ma eseguire la ricetta manualmente. Per gli aggiornamenti di pacchetti, puoi anche usare le risorse yum_package (Amazon Linux) o apt_package (Ubuntu) invece di un comando shell.

- Accedere a ogni istanza con SSH ed eseguire i comandi appropriati manualmente.

Utilizzo dei gruppi di sicurezza

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Gruppi di sicurezza

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ogni EC2 istanza Amazon ha uno o più gruppi di sicurezza associati che regolano il traffico di rete dell'istanza, proprio come un firewall. Un gruppo di sicurezza dispone di una o più regole, ciascuna delle quali specifica una determinata categoria di traffico consentito. Una regola specifica le seguenti informazioni:

- Tipo di traffico consentito, ad esempio SSH o HTTP
- Protocollo del traffico, ad esempio TCP o UDP
- Intervallo di indirizzi IP da cui ha origine il traffico
- Intervallo di porte consentite per il traffico

I gruppi di sicurezza dispongono di due tipi di regole:

- Le regole in entrata regolamentano il traffico di rete in entrata.

Ad esempio, le istanze del server di applicazioni in genere ha un regola in entrata che consente il traffico HTTP in entrata da qualsiasi indirizzo IP alla porta 80 e un'altra regola in entrata che consente il traffico SSH in entrata sulla porta 22 dal set specificato di indirizzi IP.

- Le regole in uscita regolamentano il traffico di rete in uscita.

  In genere è consigliabile utilizzare l'impostazione predefinita, che consente qualsiasi traffico in uscita.

Per ulteriori informazioni sui gruppi di sicurezza, consulta Amazon EC2 Security Groups.

La prima volta che crei uno stack in una regione, AWS OpsWorks Stacks crea un gruppo di sicurezza integrato per ogni livello con un set di regole appropriato. Tutti i gruppi hanno regole in uscita predefinite, che consentono tutto il traffico in uscita. In generale, le regole in entrata consentono quanto segue:

- Traffico TCP, UDP e ICMP in entrata dai livelli Stacks appropriati AWS OpsWorks
- Traffico TCP in entrata sulla porta 22 (accesso SSH)

> **Warning**
>
> La configurazione predefinita del gruppo di sicurezza apre la porta SSH (22) per qualsiasi posizione di rete (0.0.0.0/0). Ciò consente a tutti gli indirizzi IP di accedere all'istanza tramite SSH. Per gli ambienti di produzione, devi utilizzare una configurazione che consenta solo l'accesso SSH da un indirizzo IP o da un intervallo indirizzi specificato. Aggiornare i gruppi di sicurezza predefiniti subito dopo la creazione oppure utilizzare invece gruppi di sicurezza personalizzati.

- Per i livelli del server Web, tutto il traffico TCP e UDP in entrata verso le porte 80 (HTTP) e 443 (HTTPS)

> **Note**
>
> Il gruppo di sicurezza predefinito `AWS-OpsWorks-RDP-Server` viene assegnato a tutte le istanze Windows per consentire l'accesso RDP. Tuttavia, per impostazione predefinita, tale gruppo non ha regole. Se esegui uno stack Windows e vuoi utilizzare RDP per accedere

alle istanze, devi aggiungere una regola in entrata che consenta l'accesso RDP. Per ulteriori informazioni, consulta Accesso con RDP.

Per visualizzare i dettagli di ciascun gruppo, vai alla EC2console Amazon, seleziona Security Groups nel riquadro di navigazione e seleziona il gruppo di sicurezza del layer appropriato. Ad esempio, AWS- OpsWorks -Default-Server è il gruppo di sicurezza integrato predefinito per tutti gli stack e AWS OpsWorks - WebApp è il gruppo di sicurezza integrato predefinito per lo stack di esempio Chef 12.

> **Note**
>
> Se elimini accidentalmente un gruppo di sicurezza AWS OpsWorks Stacks, il modo migliore per ricrearlo è fare in modo che Stacks esegua l'operazione al posto tuo. AWS OpsWorks Basta creare un nuovo stack nella stessa regione AWS e il VPC, se AWS OpsWorks presente, e Stacks ricreerà automaticamente tutti i gruppi di sicurezza integrati, incluso quello che hai eliminato. È quindi possibile eliminare lo stack se non è più necessario. I gruppi di sicurezza verranno conservati. Per ricreare il gruppo di sicurezza manualmente, dovrai realizzare un duplicato esatto dell'originale, rispettando anche l'uso di maiuscole e minuscole per il nome del gruppo.
> Inoltre, AWS OpsWorks Stacks tenterà di ricreare tutti i gruppi di sicurezza integrati se si verifica una delle seguenti condizioni:
>
> - Puoi apportare eventuali modifiche alla pagina delle impostazioni dello stack nella console Stacks. AWS OpsWorks
> - Avvii una delle istanze dello stack.
> - Crei un nuovo stack.

Puoi utilizzare uno dei seguenti approcci per specificare i gruppi di sicurezza. Utilizzi l'impostazione Usa i gruppi OpsWorks di sicurezza per specificare le tue preferenze quando crei uno stack.

- Sì (impostazione predefinita): AWS OpsWorks Stacks associa automaticamente il gruppo di sicurezza integrato appropriato a ciascun livello.

  Puoi migliorare il gruppo di sicurezza integrato di un livello aggiungendo un gruppo di sicurezza personalizzato con le impostazioni desiderate. Tuttavia, quando Amazon EC2 valuta più gruppi

di sicurezza, utilizza le regole meno restrittive, quindi non puoi utilizzare questo approccio per specificare regole più restrittive rispetto al gruppo integrato.

- No, AWS OpsWorks Stacks non associa i gruppi di sicurezza integrati ai livelli.

  Devi creare gruppi di sicurezza appropriati e associarne almeno uno a ciascun livello creato. Utilizza questo approccio per specificare regole più restrittive rispetto a quelle dei gruppi integrati. È da notare che puoi ancora associare manualmente un gruppo di sicurezza integrato a un livello, se preferisci. I gruppi di sicurezza personalizzati sono necessari solo per quei livelli che necessitano di impostazioni personalizzate.

> **Important**
>
> Se utilizzi i gruppi di sicurezza integrati, non puoi creare regole più restrittive modificando manualmente le impostazioni del gruppo. Ogni volta che crei uno stack, AWS OpsWorks Stacks sovrascrive le configurazioni dei gruppi di sicurezza integrati, quindi tutte le modifiche apportate andranno perse alla successiva creazione di uno stack. Se un livello richiede impostazioni di gruppo di sicurezza più restrittive rispetto al gruppo di sicurezza integrato, imposta Usa gruppi di OpsWorks sicurezza su No, crea gruppi di sicurezza personalizzati con le tue impostazioni preferite e assegnali ai livelli al momento della creazione.

AWS OpsWorks Supporto Stacks per Chef 12 Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione fornisce una breve panoramica di AWS OpsWorks Stacks for Chef 12 Linux. Per informazioni su Chef 12 su Windows, consulta Nozioni di base: Windows. Per informazioni sulle versioni precedenti di Chef su Linux, consulta Chef 11.10 e versioni precedenti per Linux.

Panoramica

AWS OpsWorks Stacks supporta Chef 12, l'ultima versione di Chef, per gli stack Linux. Per ulteriori informazioni, consulta la pagina con le informazioni su Chef.

AWS OpsWorks Stacks continua a supportare Chef 11.10 per gli stack Linux. Tuttavia, se sei un utente Chef esperto e desideri trarre vantaggio dall'ampia selezione di libri di ricette della community oppure scrivere i tuoi libri di ricette personali, ti consigliamo di utilizzare Chef 12. Gli stack Chef 12 offrono i seguenti vantaggi rispetto agli stack Chef 11.10 e precedenti per Linux:

- Due esecuzioni Chef separate: quando un comando viene eseguito su un'istanza, l'agente AWS OpsWorks Stacks ora esegue due esecuzioni Chef isolate: una per attività che integrano l'istanza con altri servizi AWS come AWS Identity and Access Management (IAM) e una per i tuoi libri di cucina personalizzati. La prima esecuzione di Chef installa l'agente AWS OpsWorks Stacks sull'istanza ed esegue attività di sistema come l'impostazione e la gestione degli utenti, l'impostazione e la configurazione dei volumi, la configurazione delle CloudWatch metriche e così via. La seconda esecuzione è dedicata esclusivamente all'esecuzione delle ricette personalizzate per AWS OpsWorks Eventi del ciclo di vita degli stack. Questa seconda esecuzione ti consente di utilizzare i tuoi libri di ricette Chef o i libri di ricette della community.
- Risoluzione dei conflitti dello spazio dei nomi - Prima di Chef 12, AWS OpsWorks Stacks eseguiva le attività di sistema e le ricette integrate e personalizzate in un ambiente condiviso. Ciò ha provocato conflitti nello spazio dei nomi e una mancanza di chiarezza sulle ricette eseguite da Stacks. AWS OpsWorks Pertanto, sebbene si tratti di un'attività lunga e soggetta a errori, è stato necessario sovrascrivere manualmente le configurazioni predefinite indesiderate. In Chef 12 per Linux, AWS OpsWorks Stacks non supporta più i ricettari Chef integrati per ambienti server di applicazioni come PHP, Node.js o Rails. Eliminando le ricette integrate, AWS OpsWorks Stacks elimina il problema delle collisioni di denominazione tra ricette integrate e personalizzate.
- Forte supporto per i libri di cucina della community di Chef: AWS OpsWorks Stacks Chef 12 Linux offre maggiore compatibilità e supporto per i libri di cucina della community del supermercato Chef. Ora puoi utilizzare libri di cucina comunitari superiori ai libri di cucina integrati forniti in precedenza da AWS OpsWorks Stacks, libri di cucina progettati per essere utilizzati con gli ambienti e i framework di server delle applicazioni più recenti. Puoi eseguire la maggior parte di questi libri di ricette senza modifiche su Chef 12 per Linux. Per ulteriori informazioni, vai a Chef Supermarket sul sito Web Learn Chef, sul sito Web ChefSupermarket e all'archivio Chef Cookbooks su. GitHub
- Aggiornamenti tempestivi di Chef 12: AWS OpsWorks Stacks aggiornerà il suo ambiente Chef all'ultima versione di Chef 12 poco dopo ogni versione di Chef. Con Chef 12, gli aggiornamenti minori di Chef e le nuove versioni dell'agente AWS OpsWorks Stacks coincideranno. Ciò ti

consente di testare nuove versioni di Chef direttamente e consente alle tue applicazioni e alle ricette Chef di trarre vantaggio dalle ultime funzionalità di Chef.

Per ulteriori informazioni sulle versioni di Chef supportate prima di Chef 12, consulta Chef 11.10 e versioni precedenti per Linux.

Passaggio a Chef 12

Le modifiche apportate a Key AWS OpsWorks Stacks per Chef 12 Linux, rispetto al supporto per le precedenti versioni di Chef 11.10, 11.4 e 0.9, sono le seguenti:

- I livelli integrati non sono più forniti o supportati per gli stack Chef 12 per Linux. Poiché vengono eseguite solo le tue ricette personalizzate, l'eliminazione di questo supporto offre trasparenza totale in merito al modo in cui l'istanza viene configurata e semplifica la scrittura e la gestione dei libri di ricette personalizzati. Ad esempio, non è più necessario sovrascrivere gli attributi delle ricette Stacks integrate. AWS OpsWorks La rimozione dei livelli integrati consente inoltre a AWS OpsWorks Stacks di supportare meglio i libri di cucina sviluppati e gestiti dalla community di Chef, in modo da poterli sfruttare appieno. I tipi di layer integrati non più disponibili in Chef 12 per Linux sono: AWS Flow (Ruby), Ganglia HAProxy, Java App Server, Memcached, MySQL, Node.js App Server, PHP App Server, Rails AppServer e Static Web Server.
  - Poiché AWS OpsWorks Stacks esegue le ricette fornite da te, non è più necessario sovrascrivere gli attributi Stacks integrati eseguendo libri di cucina personalizzati. AWS OpsWorks Per ignorare attributi nelle tue ricette o in quelle della community, segui le istruzioni e gli esempi nella sezione relativa agli attributi della documentazione di Chef 12.
- AWS OpsWorks Stacks continua a fornire supporto per i seguenti livelli per gli stack Linux di Chef 12:
  - Livelli personalizzati
  - Amazon RDS Service Layer
  - Livelli del cluster ECS
- La configurazione dello stack e i contenitori di dati per Chef 12 Linux sono stati modificati per assomigliare ai loro omologhi per Chef 12.2 Windows. In questo modo è più semplice eseguire query, analizzare e risolvere i problemi relativi a questi contenitori di dati, soprattutto se lavori con stack che utilizzano tipi di sistemi operativi diversi. Tieni presente che AWS OpsWorks Stacks non supporta sacchetti di dati crittografati. Per archiviare i dati riservati in forma crittografata, ad esempio le password o i certificati, ti consigliamo di archiviarli in un bucket S3 privato. Potrai quindi creare una ricetta personalizzata che utilizza l'SDK Amazon per Ruby per recuperare i dati. Per

disporre di un esempio, consulta Utilizzo dell'SDK for Ruby. Per ulteriori informazioni, consulta AWS OpsWorks Riferimento a Stacks Data Bag.

- In Chef 12 Linux, Berkshelf non è più installato nelle istanze dello stack. Invece, ti consigliamo di usare Berkshelf su un computer di sviluppo locale per creare localmente il pacchetto delle dipendenze del tuo libro di ricette. Quindi carica il pacchetto, con le dipendenze incluse, su Amazon Simple Storage Service. Modifica infine lo stack di Chef 12 Linux per utilizzare il pacchetto caricato come origine del libro di ricette. Per ulteriori informazioni, consulta Creazione di pacchetti di dipendenze di libri di ricette a livello locale.
- Le configurazioni RAID per volumi EBS non sono più supportate. Per migliorare le prestazioni, puoi utilizzare IOPS forniti per Amazon Elastic Block Store (Amazon EBS).
- autofs non è più supportato.
- I repository di Subversion non sono più supportati.
- Adesso, le installazioni di pacchetti OS per livello devono essere eseguite con ricette personalizzate. Per ulteriori informazioni, consulta Installazioni pacchetto per livello.

Sistemi operativi supportati

Chef 12 supporta gli stessi sistemi operativi Linux delle versioni precedenti di Chef. Per un elenco dei tipi di sistemi operativi Linux e delle versioni che gli stack Chef 12 Linux possono utilizzare, consulta Sistemi operativi Linux.

Tipi di istanze supportati

AWS OpsWorks Stacks supporta tutti i tipi di istanze per gli stack Chef 12 Linux ad eccezione dei tipi di istanze specializzate come il calcolo in cluster HPC (High Performance Computing), la GPU per cluster e i tipi di istanze di cluster ad alta memoria.

Ulteriori informazioni

Per ulteriori informazioni su come utilizzare gli stack Chef 12 per Linux, vedi quanto segue:

- Nozioni di base: esempio

  Ti presenta AWS OpsWorks Stacks guidandoti attraverso un breve esercizio pratico con la console Stacks per creare un ambiente applicativo Node.js. AWS OpsWorks
- Nozioni di base: Linux

Ti presenta AWS OpsWorks Stacks e Chef 12 Linux guidandoti in un esercizio pratico con la console AWS OpsWorks Stacks per creare uno stack Linux Chef 12 di base che contenga un livello semplice con un'app Node.js che serve il traffico.

- Livelli personalizzati

  Fornisce indicazioni sull'aggiunta di un livello che contiene libri di ricette e ricette a uno stack Chef 12 Linux. Puoi utilizzare immediatamente libri di ricette e ricette forniti dalla community Chef o puoi crearne di personali.

- Spostamento ai contenitori di dati

  Confronta e specifica le differenze tra il JSON di istanza che viene utilizzato dagli stack Linux che eseguono Chef 11 e le versioni precedenti al JSON con Chef 12. Inoltre fornisce puntatori alla documentazione di riferimento per l'istanza Chef 12 in formato JSON.

Spostamento delle impostazioni dello stack dagli attributi ai contenitori di dati

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks offre un'ampia varietà di impostazioni dello stack alle tue ricette Chef. Queste impostazioni dello stack includono valori quali:

- Fonte del ricettario Stack URLs
- Configurazioni del volume di livello
- Nomi host delle istanze
- Nomi DNS Elastic Load Balancing
- Fonte dell'app URLs
- Nomi utente

Il riferimento alle impostazioni dello stack dalle ricette rende il codice ricetta più affidabile e meno soggetto a errori rispetto alla configurazione come hardcoded delle impostazioni dello stack direttamente nelle ricette. In questo argomento viene descritto come accedere a tali impostazioni dello stack nonché come passare dagli attributi in Chef 11.10 e versioni precedenti per Linux ai contenitori di dati in Chef 12 Linux.

In Chef 11.10 e versioni precedenti per Linux, le impostazioni dello stack sono disponibili come attributi Chef ed è possibile accedervi tramite l'oggetto `node` Chef o attraverso la ricerca di Chef. Questi attributi sono archiviati nelle istanze AWS OpsWorks Stacks in un set di file JSON nella directory. `/var/lib/aws/opsworks/chef` Per ulteriori informazioni, consulta Attributi di configurazione e distribuzione dello stack: Linux.

In Chef 12 Linux, le impostazioni dello stack sono disponibili come contenitori di dati Chef e sono accessibili solo attraverso la ricerca di Chef. I data bag sono archiviati su istanze AWS OpsWorks Stacks in un set di file JSON nella `/var/chef/runs/`*`run-ID`*`/data_bags` directory, dove si *`run-ID`* trova un ID univoco che AWS OpsWorks Stacks assegna a ogni esecuzione di Chef su un'istanza. Le impostazioni dello stack non sono più disponibili come attributi Chef, pertanto non è più possibile accedervi tramite l'oggetto `node` Chef. Per ulteriori informazioni, consulta la AWS OpsWorks Riferimento a Stacks Data Bag.

Ad esempio, in Chef 11.10 e versioni precedenti per Linux, il seguente codice ricetta usa l'oggetto `node` Chef per ottenere attributi che rappresentano il nome breve e l'URL di origine di un'app. Viene quindi utilizzato il log di Chef per scrivere questi due valori di attributo:

```
Chef::Log.info ("********** The app's short name is '#{node['opsworks']
['applications'].first['slug_name']}' **********")
Chef::Log.info("********** The app's URL is '#{node['deploy']['simplephpapp']['scm']
['repository']}' **********")
```

In Chef 12 Linux, il seguente codice ricetta utilizza l'indice di ricerca `aws_opsworks_app` per ottenere i contenuti del primo elemento del contenitore di dati nel contenitore di dati `aws_opsworks_app`. Il codice, quindi, scrive due messaggi nel log di Chef, uno con il contenuto del contenitore di dati del nome breve dell'app e l'altro con il contenuto del contenitore di dati dell'URL di origine dell'app:

```
app = search("aws_opsworks_app").first

Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}' **********")
```

Per migrare il codice ricetta che accede alle impostazioni dello stack da Chef 11.10 e versioni precedenti per Linux a Chef 12 Linux, è necessario modificare il codice per:

- Accedere ai contenitori di dati Chef anziché agli attributi Chef.
- Utilizzare la ricerca di Chef anziché l'oggetto `node` Chef.
- Usa i nomi dei data bag di AWS OpsWorks Stacks come`aws_opsworks_app`, invece di usare nomi di attributi AWS OpsWorks Stacks come e. `opsworks` `deploy`

Per ulteriori informazioni, consulta la AWS OpsWorks Riferimento a Stacks Data Bag.

Support per le versioni precedenti di Chef in AWS OpsWorks Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione fornisce una breve panoramica della documentazione di AWS OpsWorks Stacks per le versioni precedenti di Chef.

Chef 11.10 e versioni precedenti per Linux

Fornisce documentazione sul supporto di AWS OpsWorks Stacks per Chef 11.10, 11.4 e 0.9 per gli stack Linux.

Chef 11.10 e versioni precedenti per Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione fornisce una breve panoramica della documentazione di AWS OpsWorks Stacks per Chef 11.10, 11.4 e 0.9 per Linux.

Nozioni di base sugli stack Linux Chef 11

Fornisce una procedura guidata che illustra come creare uno stack del server dell'applicazione PHP semplice ma funzionale.

Creazione del primo stack Node.js

Descrive come creare uno stack di Linux che supporti un server dell'applicazione Node.js e come distribuire un'applicazione semplice.

Personalizzazione degli stack AWS OpsWorks

Descrive come personalizzare AWS OpsWorks Stacks per soddisfare requisiti specifici.

Introduzione ai libri di ricette

Descrive come implementare le ricette per le AWS OpsWorks istanze Stacks.

Bilanciamento del carico di un livello

Descrive come utilizzare le opzioni di bilanciamento del carico AWS OpsWorks Stacks disponibili.

Esecuzione di uno stack in un VPC

Descrive come creare ed eseguire uno stack in un virtual private cloud.

Migrazione da Chef Server

Fornisce linee guida per la migrazione da Chef Server a Stacks. AWS OpsWorks

AWS OpsWorks Stacks Layer Reference

Descrive i layer integrati di AWS OpsWorks Stacks disponibili.

Componenti Cookbook

Descrive i tre componenti standard del libro di ricette: attributi, modelli e ricette.

[Attributi di configurazione e distribuzione dello stack: Linux]

Descrive la configurazione dello stack e gli attributi di distribuzione per Linux.

[Attributi dei libri di ricette predefiniti]

Descrive come utilizzare gli attributi della ricetta integrata per controllare la configurazione del software installato.

[Risoluzione dei problemi di Chef 11.10 e delle versioni precedenti per Linux]

Descrive gli approcci per la risoluzione di vari problemi in AWS OpsWorks Stacks.

Nozioni di base sugli stack Linux Chef 11

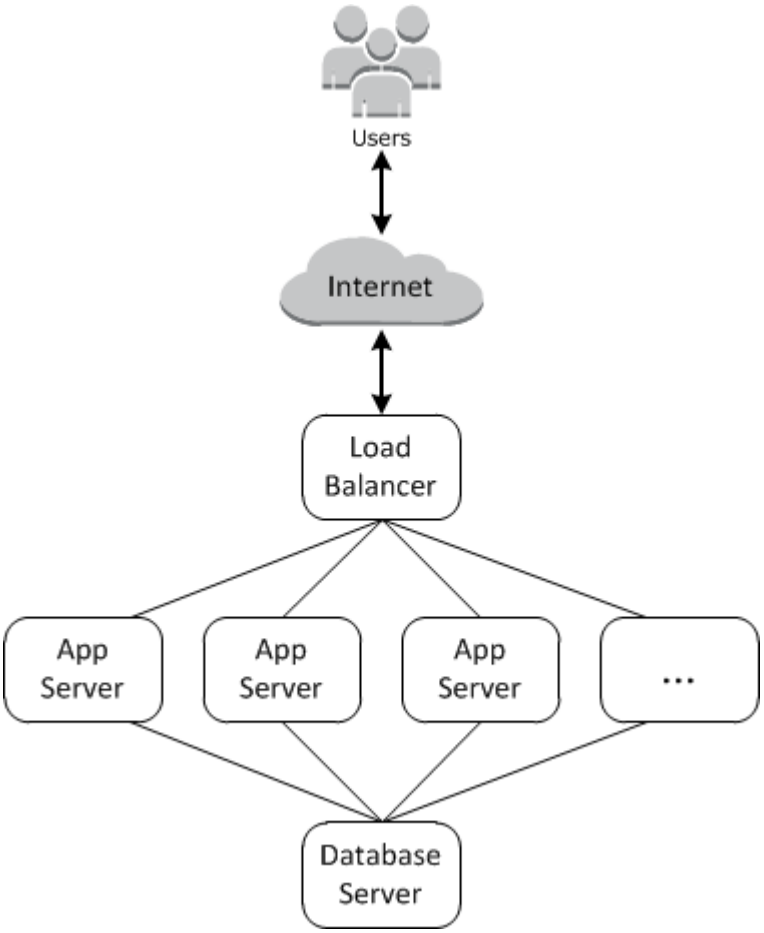
Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Note

Questa sezione descrive come iniziare a utilizzare gli stack Linux con Chef 11. Per informazioni sulle operazioni di base con gli stack Linux Chef 12, consulta Nozioni di base: Linux. Per informazioni sulle operazioni di base con gli stack Windows Chef 12, consulta Nozioni di base: Windows.

Le applicazioni basate sul cloud di solito richiedono un gruppo di risorse correlate (server di applicazioni, server di database e così via) che devono essere create e gestite collettivamente. Questa raccolta di istanze viene chiamata stack. Un semplice stack di applicazioni potrebbe essere simile al seguente.



L'architettura di base è composta dagli elementi seguenti:

- Un sistema di bilanciamento del carico per distribuire il traffico generato dagli utenti in entrata in modo uniforme in tutti i server di applicazioni.
- Un set di istanze di server di applicazioni, in numero sufficiente a gestire il traffico.
- Un server di database per fornire ai server di applicazioni un datastore back-end.

Inoltre, solitamente è necessario un modo per distribuire le applicazioni ai server di applicazioni, monitorare lo stack e così via.

AWS OpsWorks Stacks offre un modo semplice e diretto per creare e gestire gli stack e le applicazioni e le risorse associate. Questo capitolo introduce le nozioni di base di AWS OpsWorks Stacks, insieme ad alcune delle sue funzionalità più sofisticate, illustrando il processo di creazione dello stack di application server illustrato nel diagramma. Utilizza un modello di sviluppo incrementale che AWS OpsWorks Stacks rende facile da seguire: configura uno stack di base e, una volta che funziona correttamente, aggiungi componenti fino ad arrivare a un'implementazione completa.

- Fase 1: completamento dei prerequisiti mostra come ottenere la configurazione per iniziare la procedura guidata.
- Fase 2: Creazione di uno stack di server di applicazioni semplice - Chef 11 mostra come creare uno stack minimo costituito da un singolo server di applicazioni.
- Fase 3. Aggiunta di un archivio dati back-end mostra come aggiungere un server di database e collegarlo al server di applicazioni.
- Fase 4: Scalare Out MyStack mostra come scalare uno stack per gestire l'aumento di carico aggiungendo più server di applicazioni e un sistema di bilanciamento del carico per distribuire il traffico in entrata.

Argomenti

- Fase 1: completamento dei prerequisiti
- Fase 2: Creazione di uno stack di server di applicazioni semplice - Chef 11
- Fase 3. Aggiunta di un archivio dati back-end
- Fase 4: Scalare Out MyStack
- Fase 5: Eliminare MyStack

Fase 1: completamento dei prerequisiti

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Prima di poter avviare la procedura guidata, completa le seguenti fasi di configurazione. Questi passaggi di configurazione includono la registrazione di un AWS account, la creazione di un utente amministrativo e l'assegnazione delle autorizzazioni di accesso a Stacks. AWS OpsWorks

Se hai già completato una delle procedure guidate di Guida introduttiva a AWS OpsWorks Stacks, i prerequisiti di questa procedura guidata sono già soddisfatti; puoi passare direttamente a Fase 2: Creazione di uno stack di server di applicazioni semplice - Chef 11.

Argomenti

- Registrati per un Account AWS
- Crea un utente con accesso amministrativo
- Assegna le autorizzazioni di accesso al servizio al tuo utente

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Parte della procedura di registrazione prevede la ricezione di una telefonata o di un messaggio di testo e l'immissione di un codice di verifica sulla tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Assegna le autorizzazioni di accesso al servizio al tuo utente

Abilita l'accesso al servizio AWS OpsWorks Stacks (e ai servizi correlati su cui si basa AWS OpsWorks Stacks) aggiungendo le `AmazonS3FullAccess` autorizzazioni `AWSOpsWorks_FullAccess` e al tuo ruolo o utente.

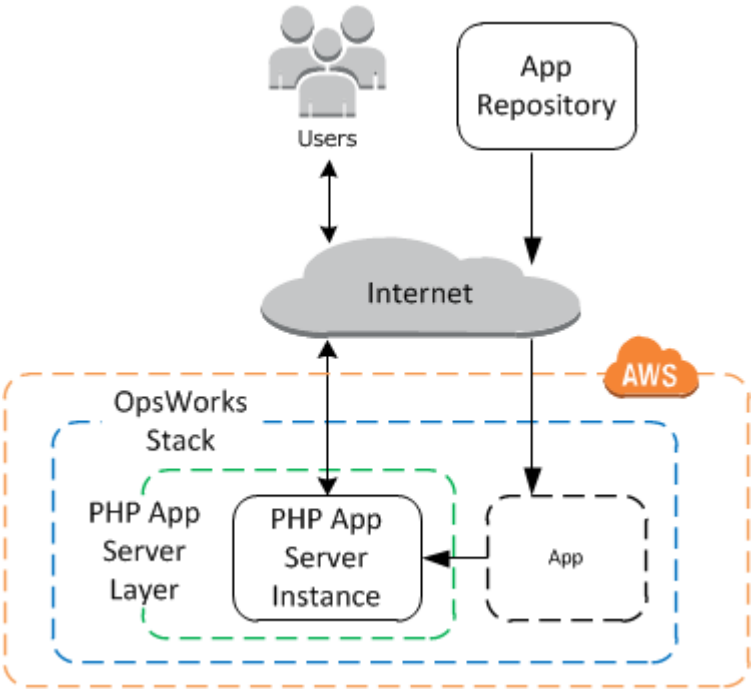
Per ulteriori informazioni sull'aggiunta di autorizzazioni, consulta Aggiungere i permessi di identità IAM (console).

A questo punto hai completato tutte le fasi di configurazione. Ora puoi avviare questa procedura guidata.

Fase 2: Creazione di uno stack di server di applicazioni semplice - Chef 11

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Uno stack server di applicazioni di base è composto da una singola istanza server di applicazioni e da un indirizzo IP pubblico per ricevere le richieste utente. Il codice dell'applicazione e dei file correlati vengono depositati in un archivio separato e distribuiti da lì al server. Il diagramma seguente illustra tale stack.



Lo stack include i componenti seguenti:

- Un livello, che rappresenta un gruppo di istanze e specifica come devono essere configurate.

  Il layer in questo esempio rappresenta un gruppo di istanze di PHP App Server.

- Un'istanza, che rappresenta un' EC2 istanza Amazon.

  In questo caso, l'istanza è configurata per eseguire un server di applicazioni PHP. I livelli possono avere un numero qualsiasi di istanze. AWS OpsWorks Stacks supporta anche diversi altri server di app. Per ulteriori informazioni, consulta Livelli del server di applicazioni.

- Un'app, che contiene le informazioni necessarie per installare un'applicazione sul server di applicazioni.

  Il codice viene archiviato in un repository remoto, ad esempio un repository Git o un bucket Amazon S3.

Le sezioni seguenti descrivono come utilizzare la console AWS OpsWorks Stacks per creare lo stack e distribuire l'applicazione. Puoi anche utilizzare un AWS CloudFormation modello per effettuare il provisioning di uno stack. Per un modello di esempio che esegue il provisioning dello stack descritto in questo argomento, consulta AWS OpsWorks Snippets.

Argomenti

- Fase 2.1: Creazione di uno stack - Chef 11
- Passaggio 2.2: Aggiungi un PHP App Server Layer - Chef 11
- Passaggio 2.3: Aggiungere un'istanza al PHP App Server Layer - Chef 11
- Fase 2.4: Creazione e distribuzione di un'app - Chef 11

Fase 2.1: Creazione di uno stack - Chef 11

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi avviare un progetto AWS OpsWorks Stacks creando uno stack, che funge da contenitore per le tue istanze e altre risorse. La configurazione dello stack specifica alcune impostazioni di base, ad esempio la regione AWS e il sistema operativo predefinito, che sono condivisi da tutte le istanze dello stack.

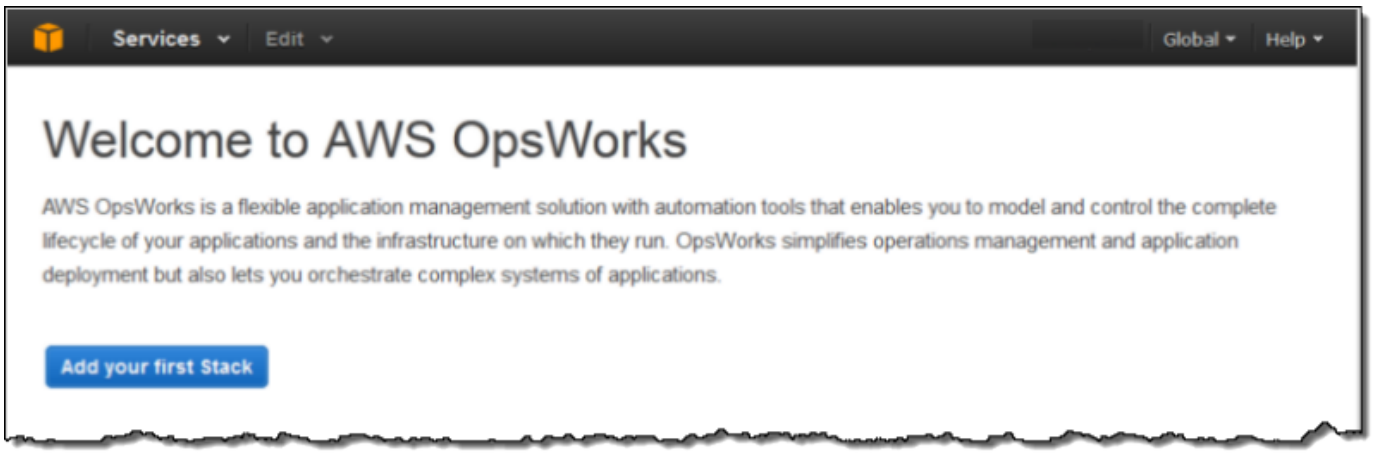
> **Note**
>
> Questa pagina ti aiuta a creare gli stack Chef 11. Per informazioni su come creare stack Chef 12, consulta l'argomento relativo alla creazione di uno stack.

Questa pagina ti aiuta a creare gli stack in Chef 11.

Per creare un nuovo stack

1. Aggiungi uno stack

   Accedere alla console AWS OpsWorks Stacks. Se l'account non ha stack esistenti, vedrai la OpsWorks pagina di benvenuto in AWS; fai clic su Aggiungi il tuo primo stack. Altrimenti, vedrai la dashboard AWS OpsWorks Stacks, che elenca gli stack del tuo account; fai clic su Aggiungi stack.



2. Configura lo stack

   Nella pagina Add Stack (Aggiungi stack), scegliere Chef 11 stack (Stack Chef 11) e specificare le seguenti impostazioni:

   Stack name (Nome stack)

   Inserisci un nome per lo stack, che può contenere caratteri alfanumerici (a—z, A—Z e 0—9) e trattini (-). Il nome dello stack di esempio utilizzato in questa procedura guidata è **`MyStack`**.

   Regione

   Seleziona Stati Uniti occidentali (Oregon) come regione dello stack.

   Accettare i valori predefiniti per le altre impostazioni, quindi fare clic su Add Stack (Aggiungi stack). Per ulteriori informazioni sulle varie impostazioni dello stack, consulta Creare un nuovo stack.

Passaggio 2.2: Aggiungi un PHP App Server Layer - Chef 11

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Anche se uno stack è fondamentalmente un contenitore di istanze, non aggiungere istanze direttamente a uno stack. Puoi aggiungere un livello, che rappresenta un gruppo di istanze correlate, e quindi aggiungere istanze al livello.

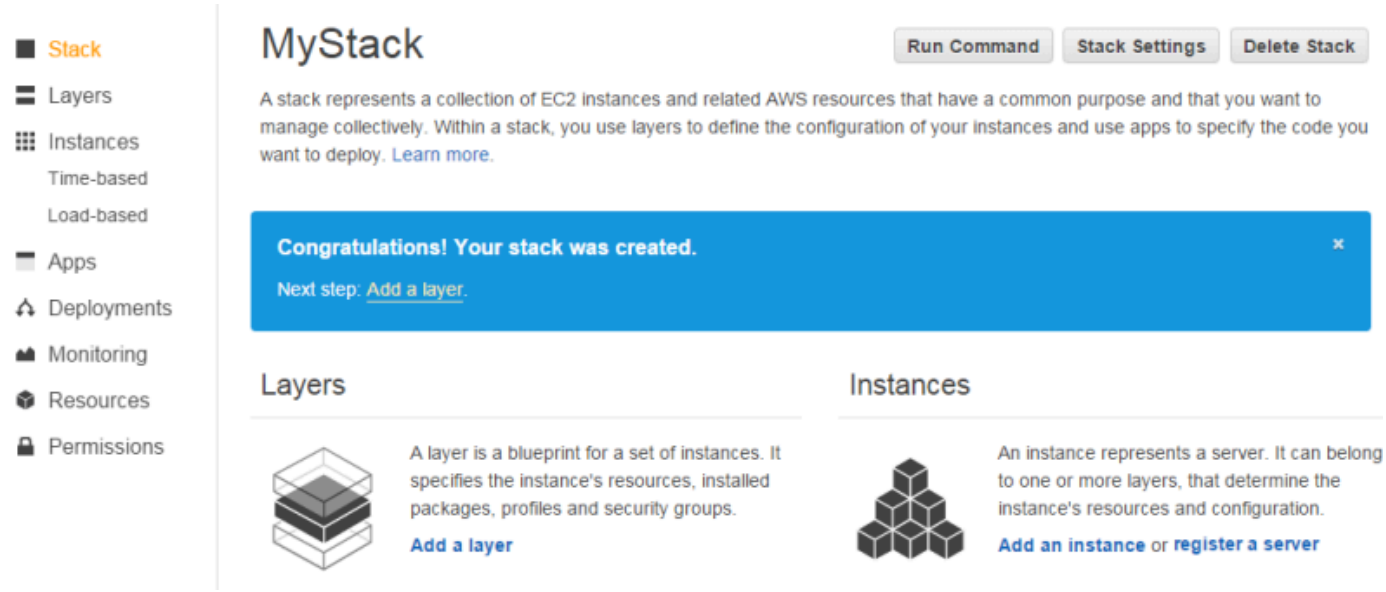
Un layer è fondamentalmente un modello che AWS OpsWorks Stacks utilizza per creare set di EC2 istanze Amazon con la stessa configurazione. Puoi aggiungere un livello allo stack per ciascun gruppo di istanze correlate. AWS OpsWorks Stacks include un set di livelli integrati per rappresentare gruppi di istanze che eseguono pacchetti software standard come un server di database MySQL o un server di applicazioni PHP. Puoi inoltre creare livelli parzialmente o interamente personalizzati in base alle specifiche esigenze. Per ulteriori informazioni, consulta Personalizzazione degli stack AWS OpsWorks.

MyStack ha un livello, il livello PHP App Server integrato, che rappresenta un gruppo di istanze che funzionano come server di applicazioni PHP. Per ulteriori informazioni, incluse le descrizioni dei livelli predefiniti, consulta Livelli.

Per aggiungere un livello PHP App Server a MyStack

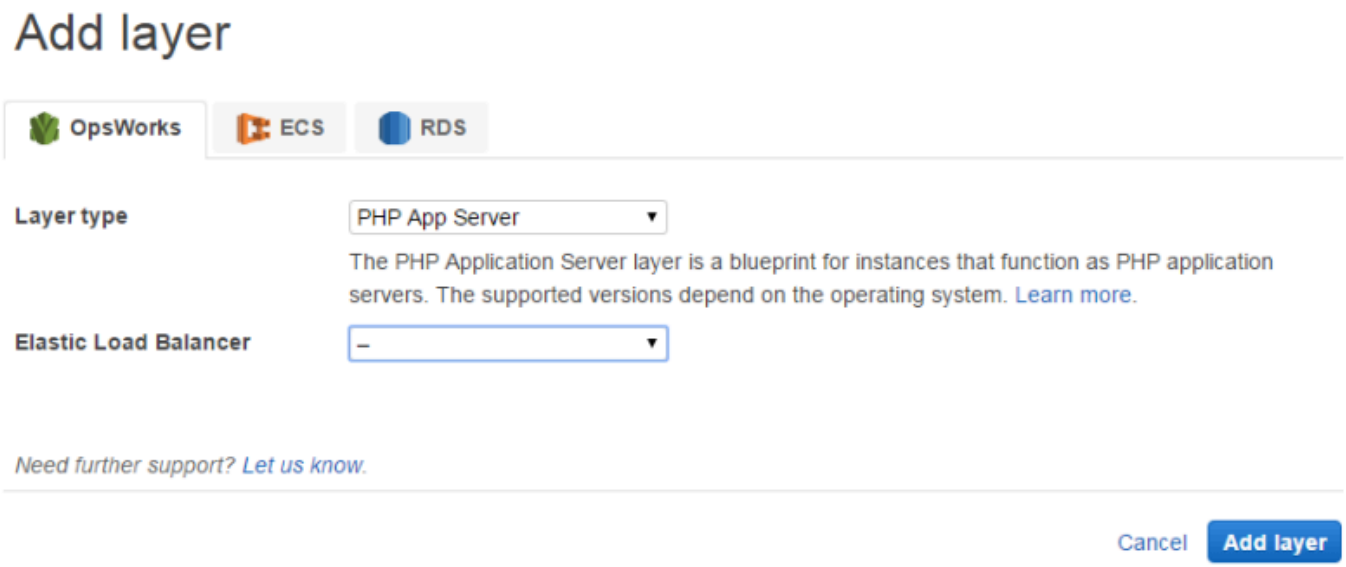
1. Aprire la pagina Add Layer (Aggiungi livello)

   Dopo aver finito di creare lo stack, AWS OpsWorks Stacks visualizza la pagina Stack. Fare clic su Add a layer (Aggiungi un livello) per aggiungere il primo livello.



2. Specificare un tipo di livello e configurare il livello

   Nella casella Tipo di layer, seleziona PHP App Server, accetta l'impostazione predefinita di Elastic Load Balancer e fai clic su Aggiungi livello. Dopo aver creato il livello, è possibile specificare altri attributi, ad esempio la configurazione del volume EBS, modificando il livello.



Passaggio 2.3: Aggiungere un'istanza al PHP App Server Layer - Chef 11

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Un'istanza AWS OpsWorks Stacks rappresenta una particolare EC2 istanza Amazon:

- La configurazione dell'istanza specifica alcuni elementi di base come il EC2operating sistema e le dimensioni di Amazon; funziona ma non fa molto.
- Il livello dell'istanza aggiunge funzionalità all'istanza stessa determinando i pacchetti da installare, se l'istanza dispone di un indirizzo IP elastico e così via.

AWS OpsWorks Stacks installa un agente su ogni istanza che interagisce con il servizio. Per aggiungere la funzionalità di un livello a un'istanza, AWS OpsWorks Stacks indirizza l'agente all'esecuzione di piccole applicazioni chiamate Chef recipes, che possono installare applicazioni e pacchetti, creare file di configurazione e così via. AWS OpsWorks Stacks esegue le ricette nei punti chiave del ciclo di vita dell'istanza. Ad esempio, OpsWorks esegue le ricette di installazione al termine dell'avvio dell'istanza per gestire attività come l'installazione del software ed esegue le ricette di Deploy quando si distribuisce un'app per installare il codice e i file correlati.

> **Note**
>
> Se sei curioso di sapere come funzionano le ricette, tutte le ricette integrate in AWS OpsWorks Stacks si trovano in un archivio pubblico GitHub : Cookbooks. OpsWorks Puoi anche creare ricette personalizzate e configurare AWS OpsWorks Stacks in modo che le esegua, come descritto più avanti.

Per aggiungere un server di applicazioni PHP MyStack, aggiungi un'istanza al livello PHP App Server che hai creato nel passaggio precedente.

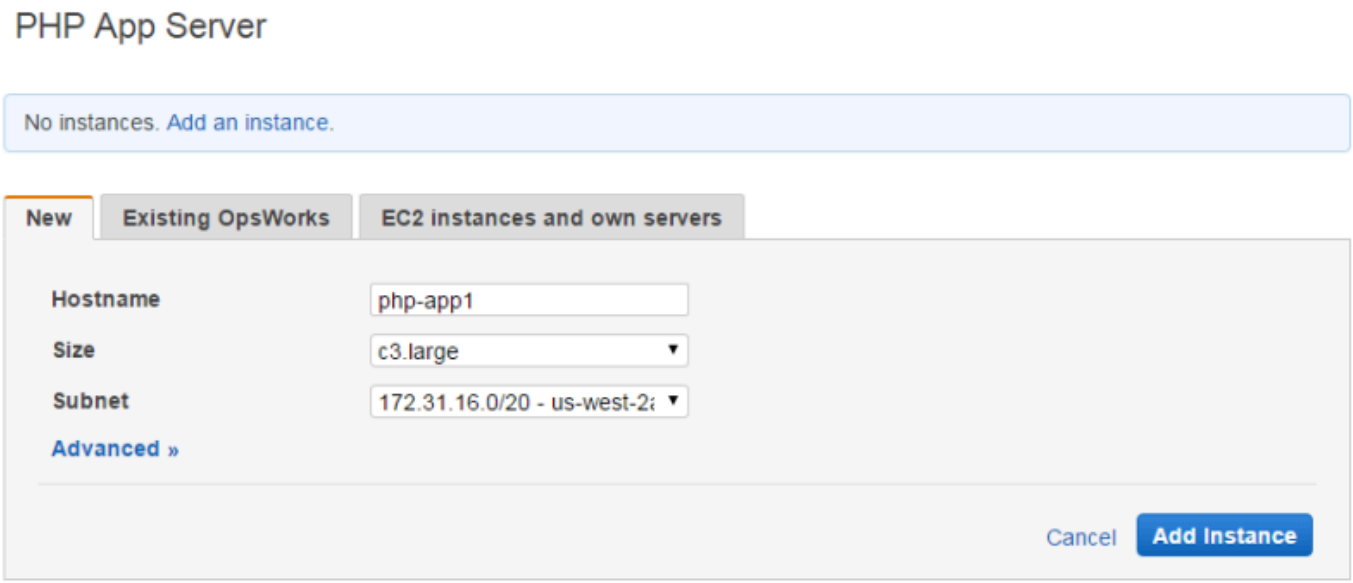
Per aggiungere un'istanza al livello PHP App Server

1. Aggiungere un'istanza

   Dopo aver finito di aggiungere il layer, AWS OpsWorks Stacks visualizza la pagina Layers. Fai clic su Istanze nel riquadro di navigazione e in PHP App Server, fai clic su Aggiungi un'istanza.

2. Configurare l'istanza

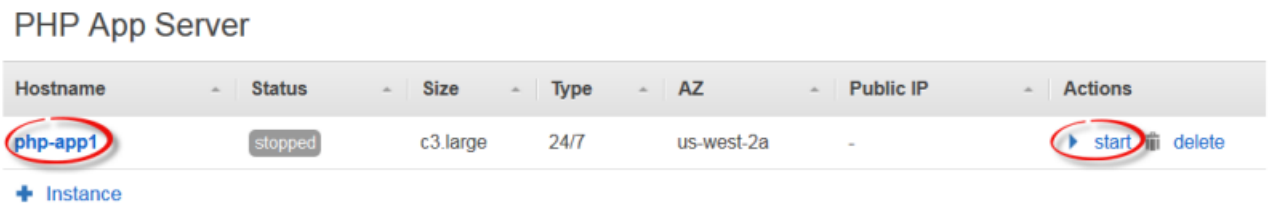
   Ogni istanza ha un nome host predefinito generato per te da AWS OpsWorks Stacks. In questo esempio, AWS OpsWorks Stacks aggiunge semplicemente un numero al nome breve del layer. Puoi configurare ogni istanza separatamente, ad esempio sostituendo alcune delle impostazioni predefinite specificate durante la creazione dello stack (zona di disponibilità o sistema operativo). Per questa procedura guidata, accetta le impostazioni predefinite e fai clic su Add Instance (Aggiungi istanza) per aggiungere l'istanza al livello. Per ulteriori informazioni, consulta Istanze.



3. Avviare l'istanza

Hai appena specificato la configurazione dell'istanza. Devi avviare un'istanza per creare un' EC2 istanza Amazon in esecuzione. AWS OpsWorks Stacks utilizza quindi le impostazioni di configurazione per avviare un' EC2 istanza Amazon nella zona di disponibilità specificata. I dettagli relativi alla modalità di avvio dell'istanza dipendono dal tipo di dimensionamento dell'istanza. Nella fase precedente hai creato un'istanza con il tipo di dimensionamento predefinito 24/7. Tale tipo di istanza deve essere avviato manualmente e rimarrà in esecuzione finché non viene arrestata manualmente. Puoi anche creare tipi di scalabilità basati sul tempo e sul carico, che AWS OpsWorks Stacks avvia e interrompe automaticamente in base a una pianificazione o al carico corrente. Per ulteriori informazioni, consulta Gestione del carico con istanze basate sul tempo e sul carico.

Vai a php-app1 sotto PHP App Server e fai clic su start nella colonna Azioni della riga per avviare l'istanza.

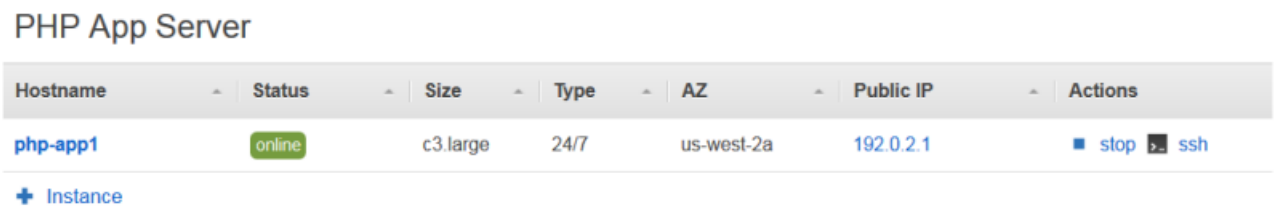


4. Monitorare lo stato dell'istanza durante l'avvio

   In genere sono necessari alcuni minuti per avviare l' EC2 istanza Amazon e installare i pacchetti. Durante il processo di avvio, nel campo Status (Stato) dell'istanza viene visualizzata la seguente serie di valori:

   1. richiesto: AWS OpsWorks Stacks ha chiamato il EC2 servizio Amazon per creare l' EC2 istanza Amazon.
   2. in sospeso: AWS OpsWorks Stacks è in attesa dell'avvio dell' EC2istanza Amazon.
   3. avvio: l' EC2 istanza Amazon si sta avviando.
   4. running_setup: l'agente AWS OpsWorks Stacks esegue le ricette di installazione del layer, che gestiscono attività come la configurazione e l'installazione dei pacchetti, e le ricette Deploy, che distribuiscono qualsiasi app sull'istanza.
   5. online: l'istanza è pronta per l'uso.

   Non appena php-app1 è online, la pagina Instances (Istanze) dovrebbe essere simile a quella illustrata nella seguente figura:



   Nella parte superiore della pagina è visualizzato un beve riepilogo di tutte le istanze dello stack. Nell'esempio è visualizzata solo un'istanza online. Nella colonna Actions (Operazioni) di php-app1 si noti che la voce stop (arresto), che consente di arrestare l'istanza, ora è stata sostituita da start (avvio) e delete (eliminazione).

Fase 2.4: Creazione e distribuzione di un'app - Chef 11

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per renderla MyStack più utile, devi distribuire un'app nell'istanza PHP App Server. Puoi archiviare il codice di un'app e i file correlati in un archivio, ad esempio un archivio Git. Devi eseguire un paio di fasi per aggiungere tali file ai server di applicazioni:

> **Note**
>
> La procedura in questa sezione è valida per gli stack Chef 11. Per ulteriori informazioni su come aggiungere app ai livelli negli stack Chef 12, consulta Aggiunta di app.

1. Creare un'app.

   Un'app contiene le informazioni di cui AWS OpsWorks Stacks ha bisogno per scaricare il codice e i file correlati dal repository. Puoi anche specificare informazioni aggiuntive, ad esempio il dominio dell'app.

2. Distribuire l'app sui server di applicazioni.

   Quando distribuisci un'app, AWS OpsWorks Stacks attiva un evento del ciclo di vita Deploy. L'agente quindi esegue le ricette di ditribuzione (Distribuzione) dell'istanza, che scaricano i file nella directory appropriata, e le attività correlate, ad esempio la configurazione del server, il riavvio del servizio e così via.

> **Note**
>
> Quando crei una nuova istanza, AWS OpsWorks Stacks distribuisce automaticamente tutte le app esistenti sull'istanza. Tuttavia, quando crei una nuova app o ne aggiorni una esistente, devi distribuire l'app o aggiornare tutte le istanze esistenti manualmente.

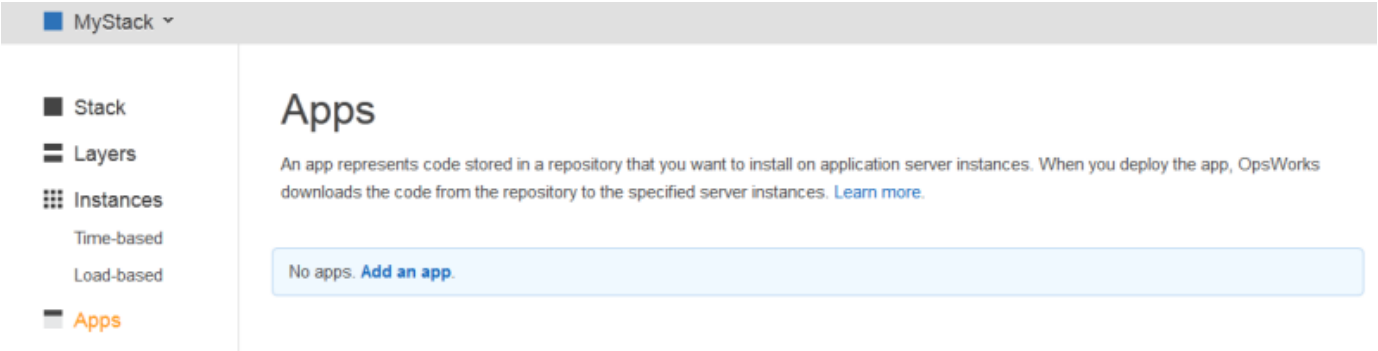
Questa fase descrive come distribuire manualmente un'app di esempio da un archivio Git pubblico a un server di applicazioni. Se desideri esaminare l'applicazione, vai su -app. https://github.com/amazonwebservices/ opsworks-demo-php-simple L'applicazione utilizzata in questo esempio si trova

nel ramo version1. AWS OpsWorks Stacks supporta anche diversi altri tipi di repository. Per ulteriori informazioni, consulta Origine dell'applicazione.

Per creare e distribuire un'app

1. Aprire la pagina Apps (App)

   Nel riquadro di navigazione, fare clic su Apps (App) e nella pagina Apps (App) fare clic su Add an app (Aggiungi un'app).



2. Configurare l'app

   Nella pagina App, specificare i seguenti valori:

   Nome

   Il nome dell'app, che AWS OpsWorks Stacks utilizza per scopi di visualizzazione. L'app di esempio è denominata**`SimplePHPApp`**. AWS OpsWorks Stacks genera anche un nome breve, simplephpapp per questo esempio, che viene utilizzato internamente e dalle ricette Deploy, come descritto più avanti.

   Tipo

   Tipo di app, che determina dove distribuire l'app. L'esempio utilizza PHP, che distribuisce l'app su istanze PHP App Server.

   Tipo di origine dati

   Server di database associato. In questo esempio, selezionare None (Nessuno). I server di database verranno descritti in Fase 3. Aggiunta di un archivio dati back-end.

   Repository type (Tipo di repository)

   Tipo di repository dell'app. L'app di esempio è memorizzata in un archivio Git.

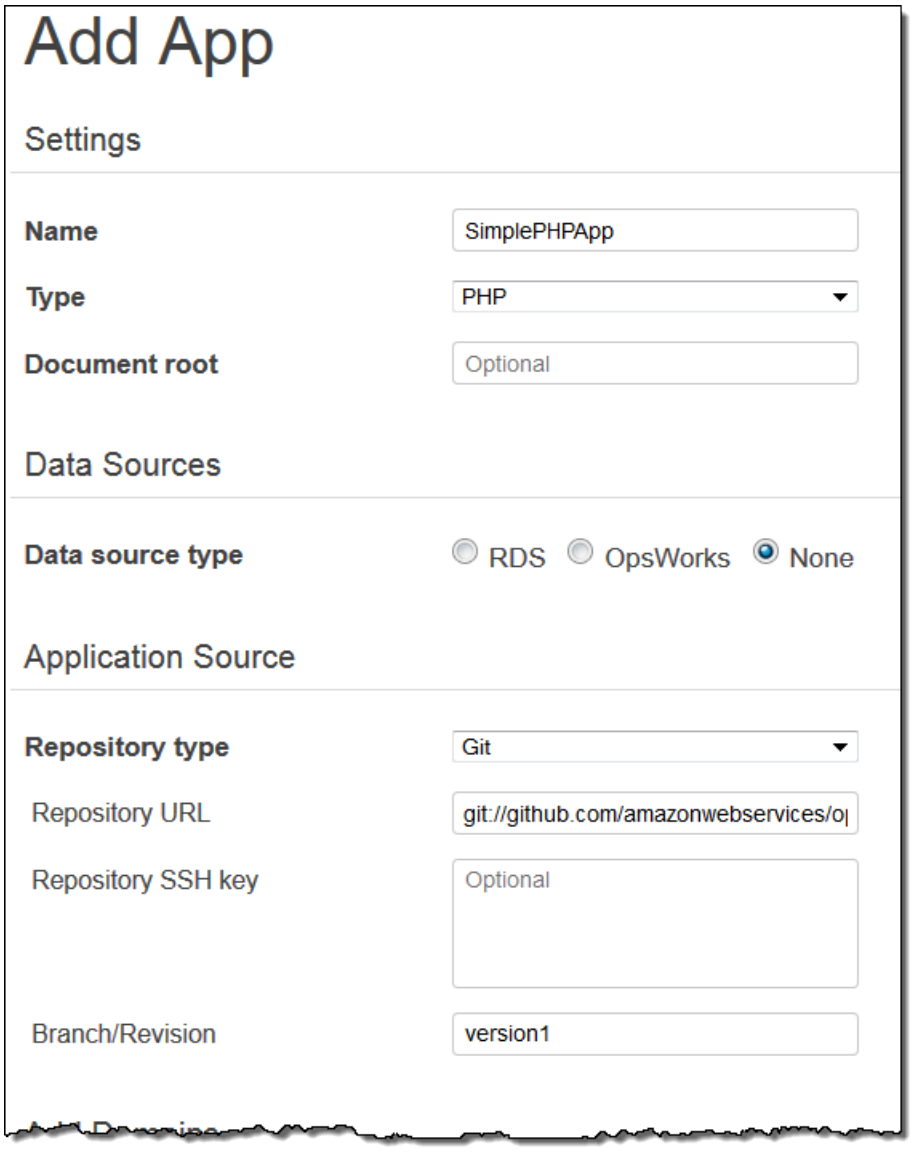
Repository URL (URL repository)

URL dell'archivio dell'app. L'esempio di URL è: **`git://github.com/awslabs/opsworks-demo-php-simple-app.git`**

Branch/Revision (Ramo/Versione)

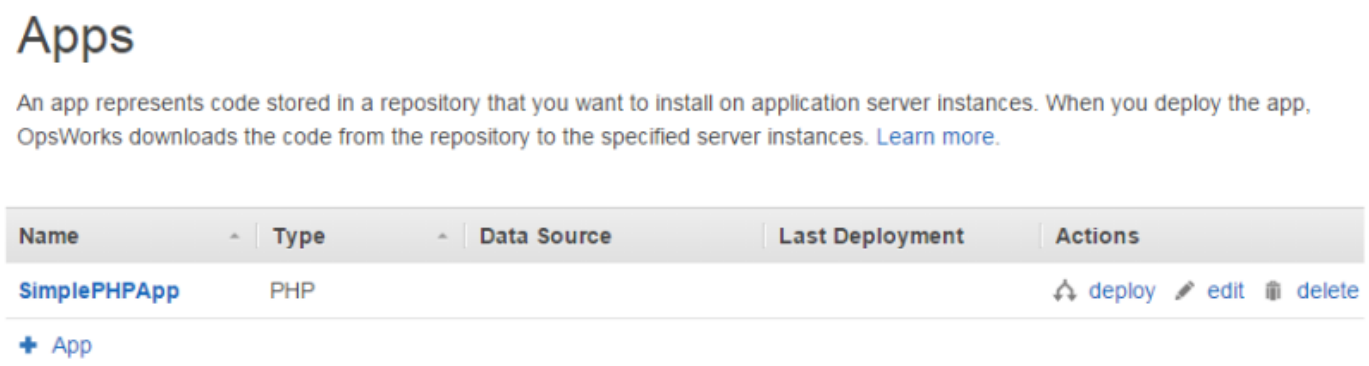
Ramo o versione dell'app. Questa parte della procedura guidata utilizza il ramo **`version1`**.

Accettare i valori predefiniti per le altre impostazioni, quindi fare clic su Add App (Aggiungi app). Per ulteriori informazioni, consulta Aggiunta di app.



3. Aprire la pagina Deployment (Distribuzione)

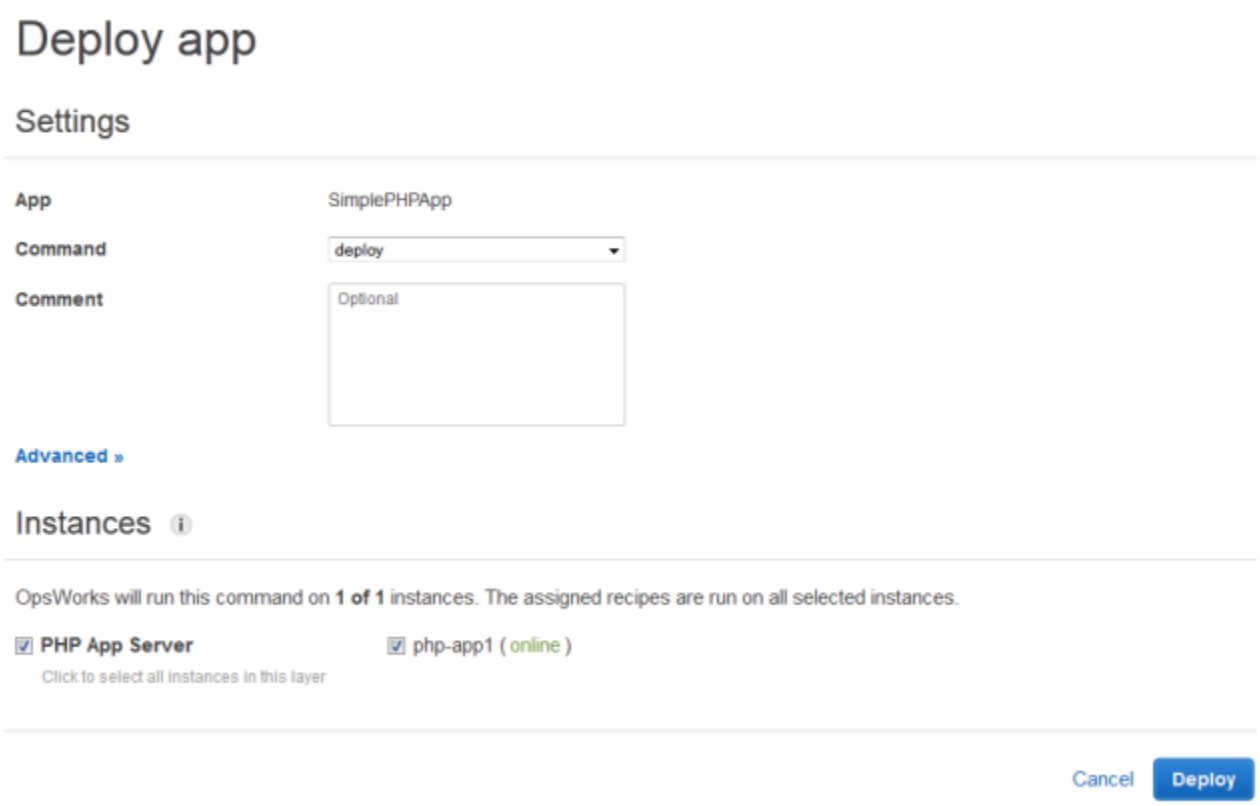
   Per installare il codice sul server, è necessario distribuire l'app. A tale scopo, fai clic su distribuisci nella colonna Azioni semplici. PHPApp



4. Distribuire l'app

   Quando distribuisci un'app, l'agente esegue le ricette Deploy sull'istanza PHP App Server, che scaricano e configurano l'applicazione.

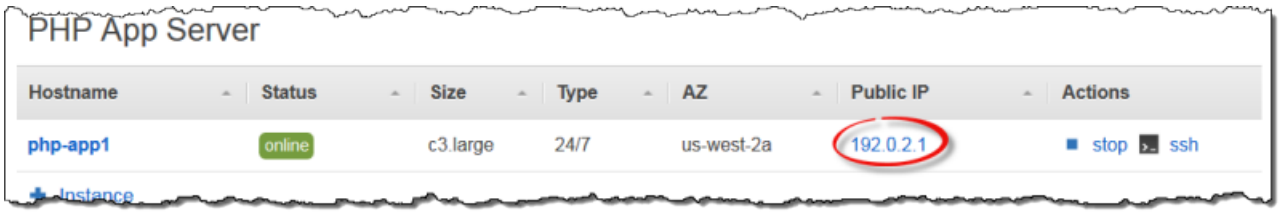
   Il campo Command (Comando) deve essere già impostato su deploy (distribuzione). Non modificare le impostazioni predefinite per le altre impostazioni e fare clic su Deploy (Distribuisci) per distribuire l'app.



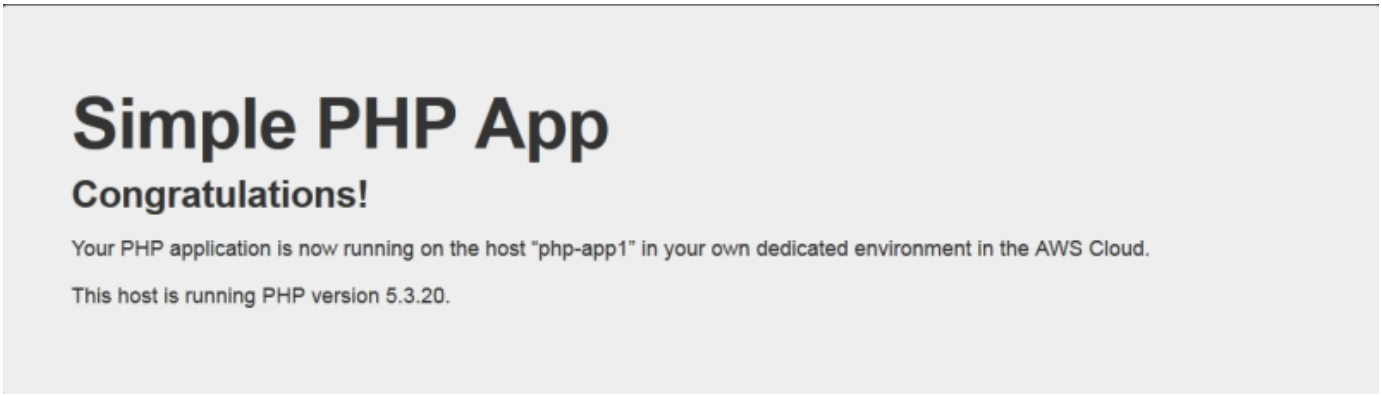
Quando la distribuzione è stata completata, nella pagina Deployment (Distribuzione) la colonna Status (Stato) visualizza la voce Successful (Operazione riuscita) e accanto a php-app1 verrà visualizzato un segno di spunta verde.

5. Esegui in modo semplice PHPApp

   Simple PHPApp è ora installato e pronto all'uso. Per eseguirla, fare clic su Instances (Istanze) nel riquadro di navigazione per passare alla pagina Instances (Istanze). Fare quindi clic sull'indirizzo IP pubblico dell'istanza php-app1.



   Nel browser in uso dovrebbe venire visualizzata una pagina simile alla seguente.

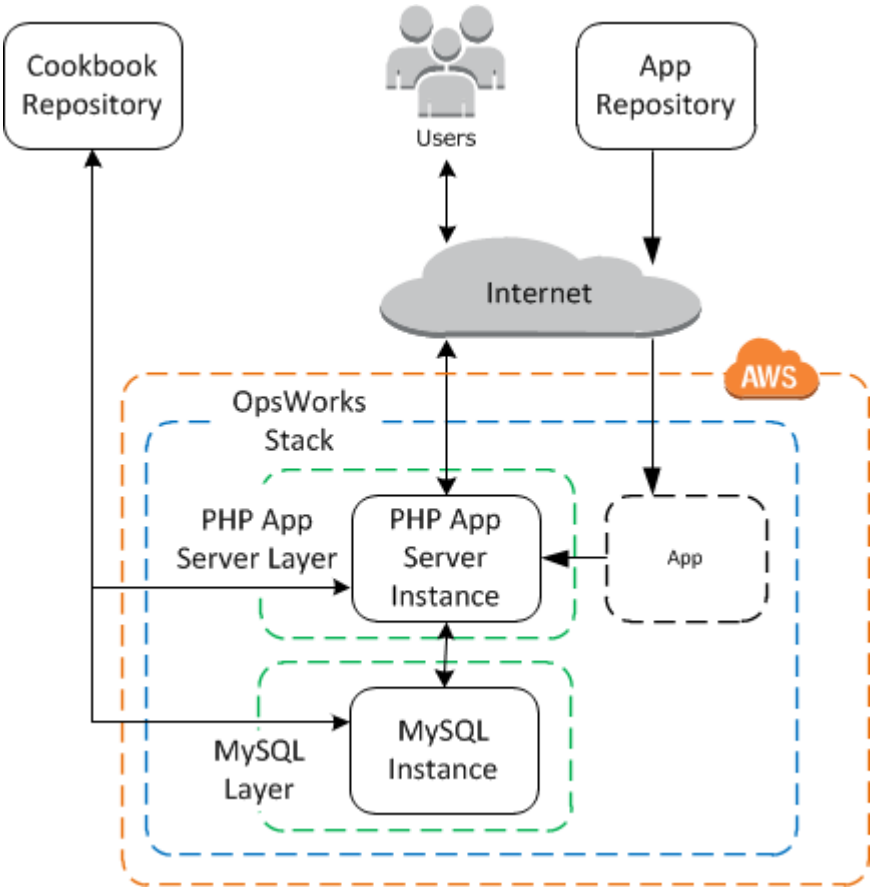


> **Note**
>
> In questa procedura guidata si presuppone che si passerà alla sezione successiva e si completerà l'intera procedura in una sessione. Se preferisci, puoi fermarti in qualsiasi momento e continuare più tardi accedendo a AWS OpsWorks Stacks e aprendo lo stack. Tuttavia, verranno addebitate le eventuali risorse AWS utilizzate, ad esempio le istanze online. Per evitare addebiti inutili, puoi interrompere l'istanza, interrompendo così l'istanza corrispondente. EC2 Sarà possibile riavviare le istanze quando si è pronti per continuare.

Fase 3. Aggiunta di un archivio dati back-end

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Nella Fase 2.1: Creazione di uno stack - Chef 11 è stato illustrato come creare uno stack che gestisce un'applicazione PHP. Tuttavia, si trattava di un'applicazione molto semplice applicazione, che si limitava a visualizzare del testo statico. Le applicazioni di produzione in genere utilizzano un archivio dati back-end, con una configurazione dello stack molto simile a quella illustrata nella figura seguente.



Questa sezione mostra come estendere MyStack per includere un server di database MySQL back-end. Tuttavia, in questa procedura non dovrai limitarti ad aggiungere un server MySQL allo stack. È inoltre necessario configurare l'app per comunicare correttamente con il server del database. AWS OpsWorks Stacks non lo fa per te; dovrai implementare alcune ricette personalizzate per gestire tale attività.

Argomenti

- Fase 3.1. Aggiunta di un database back-end
- Passaggio 3.2: Aggiornamento semplice PHPApp
- Una breve digressione: libri di cucina, ricette e attributi degli stack AWS OpsWorks
- Passaggio 3.3: Aggiungere i ricettari personalizzati a MyStack
- Fase 3.4. Esecuzione delle ricette
- Fase 3.5: Deploy Simple, versione 2 PHPApp
- Passaggio 3.6: Run Simple PHPApp

Fase 3.1. Aggiunta di un database back-end

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

La nuova versione di Simple PHPApp archivia i dati in un database di back-end. AWS OpsWorks Stacks supporta due tipi di server di database:

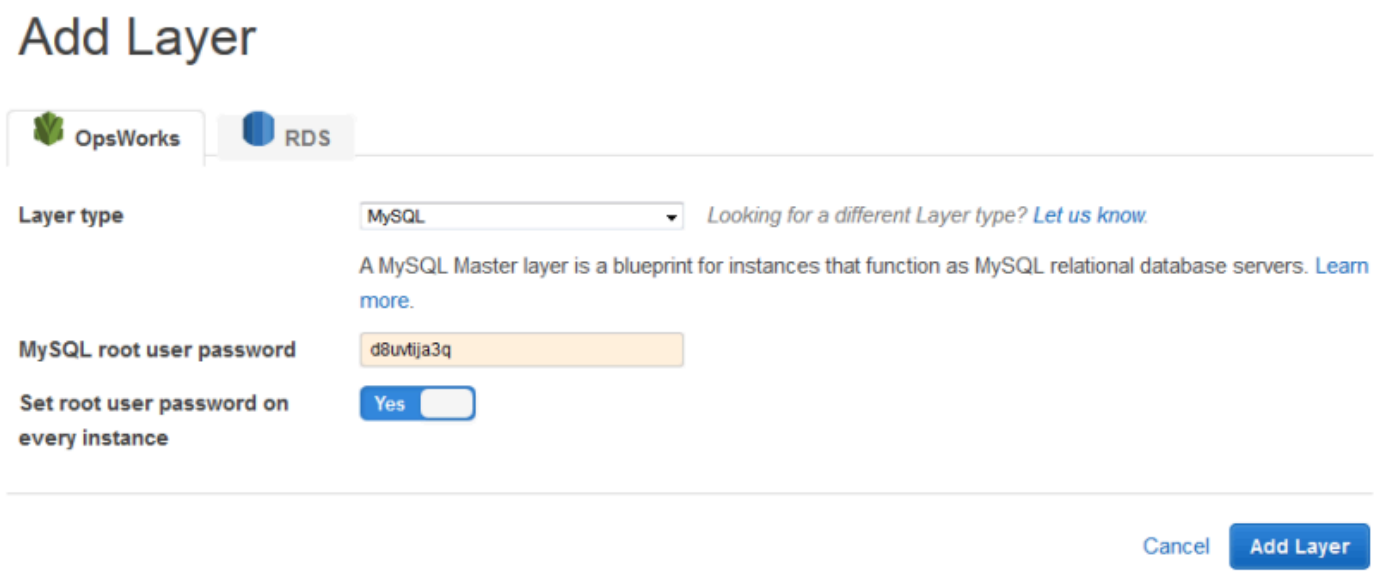
- Il livello AWS OpsWorks MySQL Stacks è un modello per la creazione di EC2 istanze Amazon che ospitano un master di database MySQL.
- Il livello di servizio Amazon RDS offre un modo per incorporare un'istanza Amazon RDS in uno stack.

Puoi anche utilizzare altri database, come Amazon DynamoDB, o creare un livello personalizzato per supportare database come MongoDB. Per ulteriori informazioni, consulta the section called “Utilizzo di un archivio dati di back-end”.

Questo esempio utilizza un livello MySQL.

Per aggiungere un livello MySQL a MyStack

1. Nella pagina Layers (Livelli), fare clic su + Layer (+ Livelli).
2. Nella pagina Add Layer (Aggiungi livello), in Layer type (Tipo di livello), selezionare MySQL, accettare le impostazioni predefinite, quindi fare clic su Add Layer (Aggiungi livello).



Per aggiungere un'istanza al livello MySQL

1. Nella pagina Layers (Livelli), nella riga MySQL fare clic su Add an instance (Aggiungi un'istanza).
2. Nella pagina Instances (Istanze), in MySQL fare clic su Add an instance (Aggiungi un'istanza).
3. Accettare le impostazioni predefinite e fare clic su Add instance (Aggiungi istanza). Non avviare l'istanza.

> **Note**
>
> AWS OpsWorks Stacks crea automaticamente un database denominato utilizzando il nome breve dell'app, simplephpapp per questo esempio. Questo nome sarà necessario se desideri usare le ricette di Chef per interagire con il database.

Passaggio 3.2: Aggiornamento semplice PHPApp

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per iniziare, è necessaria una nuova versione di Simple PHPApp che utilizzi un data store di back-end. Con AWS OpsWorks Stacks, è facile aggiornare un'applicazione. Se utilizzi un archivio Git o Subversion, puoi avere un ramo di archivio distinto per ogni versione dell'app. L'applicazione di esempio archivia una versione dell'app che utilizza un database back-end nel ramo version2 dell'archivio Git. Dovrai solo aggiornare la configurazione dell'app e specificare il nuovo ramo e quindi ridistribuire l'app.

Per aggiornare Simple PHPApp

1. Aprire la pagina di modifica dell'app

   Nel riquadro di navigazione, fai clic su App, quindi su Modifica nella colonna Azioni della PHPApp riga Semplice.
2. Aggiornare la configurazione dell'app

   Modificare le seguenti impostazioni.

   Branch/Revision (Ramo/Versione)

   Questa impostazione indica il ramo dell'archivio dell'app. La prima versione di Simple PHPApp non si connetteva a un database. Per usare una versione dell'app abilitata per il database, impostare questo valore su **`version2`**.

   Document root (Radice documento)

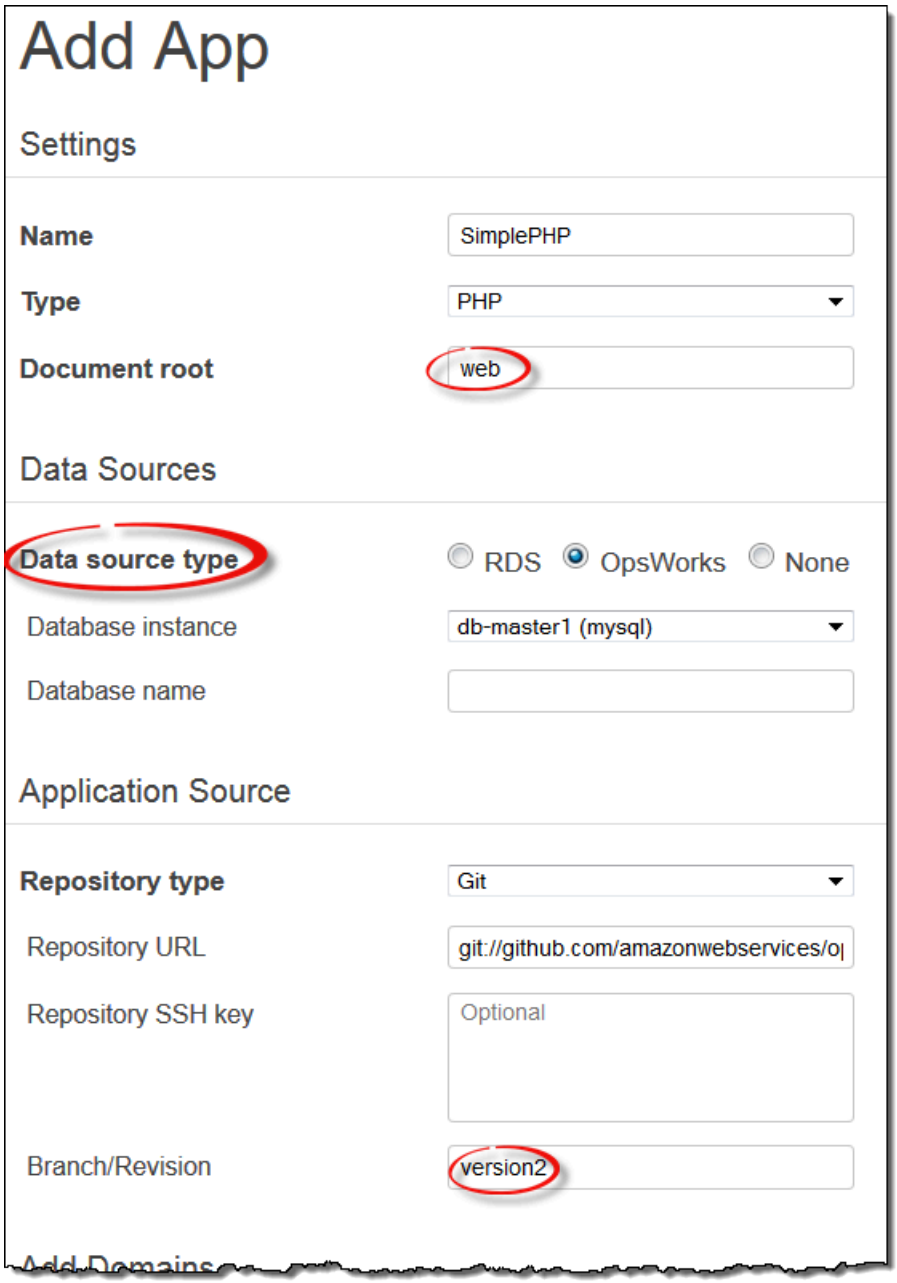
   Questa impostazione specifica la cartella radice dell'app. La prima versione di Simple PHPApp utilizzava l'impostazione predefinita, che si installa `index.php` nella cartella principale standard del server (`/srv/www`per le app PHP). Se specifichi qui una sottocartella, solo il nome, senza «/» iniziale,AWS OpsWorks Stacks la aggiunge al percorso della cartella standard. La versione 2 di Simple PHPApp dovrebbe entrare, quindi imposta Document root su. **`/srv/www/web web`**

   Data source type (Tipo di origine dati)

   Questa impostazione associa un server di database all'app. L'esempio utilizza l'istanza MySQL creata nel passaggio precedente, quindi imposta Data source type OpsWorks su e istanza Database sull'istanza creata nel passaggio precedente, db-master1 (mysql).

Lascia vuoto il nome del database; AWS OpsWorks Stacks creerà un database sul server denominato con il nome breve dell'app, simplephpapp.

Fare quindi clic su Save (Salva) per salvare la nuova configurazione.



3. Avvia l'istanza MySQL.

Dopo aver aggiornato un'app, AWS OpsWorks Stacks distribuisce automaticamente la nuova versione dell'app su tutte le nuove istanze dell'app server quando le avvii. Tuttavia, AWS OpsWorks

Stacks non distribuisce automaticamente la nuova versione dell'app sulle istanze del server esistenti; devi farlo manualmente, come descritto in. Fase 2.4: Creazione e distribuzione di un'app - Chef 11 PHPApp Ora puoi distribuire la versione aggiornata di Simple, ma per questo esempio è meglio aspettare un po'.

Una breve digressione: libri di cucina, ricette e attributi degli stack AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ora disponi di server di applicazioni e server di database, ma non sono ancora pronti per l'uso. Devi comunque configurare il database e configurare le impostazioni di connessione dell'app. AWS OpsWorks Stacks non gestisce queste attività automaticamente, ma supporta i ricettari, le ricette e gli attributi dinamici di Chef. Puoi implementare un paio di ricette, una per configurare il database e l'altra per configurare le impostazioni di connessione dell'app, e farle eseguire da AWS OpsWorks Stacks per te.

Il libro di ricette phpapp, che contiene le ricette necessarie, è già implementato ed è pronto all'uso. Puoi passare direttamente alla Passaggio 3.3: Aggiungere i ricettari personalizzati a MyStack se preferisci. Per saperne di più, questa sezione fornisce alcune informazioni di base sui libri di ricette e sulle ricette, nonché descrive il funzionamento delle ricette. Per visualizzare il libro di ricette, passa alla pagina relativa al libro di ricette phpapp.

Argomenti

- Ricette e attributi
- Configurazione del database
- Connessione dell'applicazione al database

Ricette e attributi

Una ricetta di Chef in pratica è un'applicazione Ruby specializzata che esegue attività sull'istanza quali, ad esempio, l'installazione di pacchetti, la creazione di file di configurazione, l'esecuzione di

comandi shell e così via. Gruppi di ricette correlate sono organizzati in libri di ricette, che contengono anche i file di supporto, ad esempio i modelli per la creazione dei file di configurazione.

AWS OpsWorks Stacks ha una serie di libri di cucina che supportano i livelli integrati. Puoi anche creare libri di ricette personalizzati contenenti ricette specifiche per eseguire attività personalizzate sulle istanze. Questo argomento fornisce una breve introduzione alle ricette e descrive come utilizzarle per configurare il database e le impostazioni di connessione dell'app. Per ulteriori informazioni su libri di ricette e ricette, consulta Libri di ricette e ricette o Personalizzazione degli stack AWS OpsWorks.

Le ricette in genere dipendono dagli attributi di Chef relativi ai dati di input:

- Alcuni di questi attributi sono definiti da Chef e forniscono informazioni di base sull'istanza, ad esempio il sistema operativo.
- AWS OpsWorks Stacks definisce un insieme di attributi che contengono informazioni sullo stack, come le configurazioni dei livelli, e sulle app distribuite, come l'app repository.

  Puoi aggiungere attributi personalizzati a questo set assegnando JSON personalizzato allo stack o alla distribuzione.
- I libri di ricette possono inoltre definire attributi, che sono specifici del libro di ricette interessato.

  Gli attributi del libro di ricette phpapp sono definiti in `attributes/default.rb`.

Per un elenco completo degli attributi di Stacks, consulta e. AWS OpsWorks Attributi di configurazione e distribuzione dello stack: Linux Attributi dei libri di ricette predefiniti Per ulteriori informazioni, consulta Sostituzione degli attributi.

Gli attributi sono organizzati in una struttura gerarchica, che può essere rappresentata come un oggetto JSON.

Puoi incorporare questi dati in un'applicazione utilizzando la sintassi dei nodo di Chef, come la seguente:

```
[:deploy][:simplephpapp][:database][:username]
```

Il nodo `deploy` include un singolo nodo di app, `simplephpapp`, che contiene informazioni sul database dell'app, l'archivio Git e così via. L'esempio rappresenta il valore del nome utente del database, che restituisce `root`.

Configurazione del database

Le ricette di installazione integrate nel livello MySQL creano automaticamente un database per l'app denominata con il nome breve dell'app, quindi per questo esempio hai già un database chiamato simplephpapp. Tuttavia, devi completare la configurazione mediante la creazione di una tabella in cui l'app possa archiviare i dati. È possibile creare la tabella manualmente, ma un approccio migliore consiste nell'implementare una ricetta personalizzata per gestire l'attività e lasciare che Stacks la esegua per voi. AWS OpsWorks Questa sezione descrive come la ricetta `dbsetup.rb` viene implementata. La procedura per fare in modo che AWS OpsWorks Stacks esegua la ricetta è descritta più avanti.

Per visualizzare la ricetta nell'archivio, passa a dbsetup.rb. L'esempio seguente mostra il codice di `dbsetup.rb`.

`execute` è una risorsa Chef che esegue un comando specificato. In questo caso, è un comando MySQL che crea una tabella. La direttiva `not_if` fa sì che il comando non venga eseguito se la tabella specificata esiste già. Per ulteriori informazioni sulle risorse Chef, consulta la pagina relativa alle informazioni su risorse e provider.

La ricetta inserisce i valori degli attributi nella stringa di comando, utilizzando la sintassi dei nodo descritta in precedenza. Ad esempio, il seguente consente inserisce il nome utente del database.

```
#{deploy[:database][:username]}
```

Analizziamo con maggiore dettaglio il codice:

- Per ogni iterazione, `deploy` è impostato sul nodo dell'app corrente in modo che restituisca `[:deploy][:app_name]`. In questo esempio, restituisce `[:deploy][:simplephpapp]`.
- Utilizzando i valori degli attributi di distribuzione illustrati in precedenza, l'intero nodo restituisce `root`.
- Puoi racchiudere il nodo con # {} per inserirlo in una stringa.

La maggior parte degli altri nodi vengono risolti in modo analogo. L'eccezione è `#{node[:phpapp][:dbtable]}`, che è definito dal file di attributi del libro di ricette personalizzato e restituisce il nome della tabella, ovvero `urler`. Il comando effettivo che viene eseguito sull'istanza MySQL è quindi:

```
"/usr/bin/mysql
```

```
    -uroot
    -pvjud1hw5v8
    simplephpapp
    -e'CREATE TABLE urler(
       id INT UNSIGNED NOT NULL AUTO_INCREMENT,
       author VARCHAR(63) NOT NULL,
       message TEXT,
       PRIMARY KEY (id))'
"
```

Questo comando crea una tabella denominata `urler` con i campi relativi a ID, autore e messaggio, utilizzando le credenziali e il nome di database recuperati dagli attributi di distribuzione.

Connessione dell'applicazione al database

Il secondo punto di interesse è l'applicazione, che richiede informazioni di connessione, ad esempio la password del database, per accedere alla tabella. Simple ha PHPApp effettivamente un solo file di lavoro`app.php`; tutto ciò `index.php` che fa è caricare. `app.php`

`app.php` include `db-connect.php`, che gestisce la connessione al database, ma tale file non è presente nell'archivio. Non puoi creare `db-connect.php` in anticipo perché definisce il database in base all'istanza specifica. La ricetta `appsetup.rb` genera invece `db-connect.php` utilizzando i dati di connessione ricavati dagli attributi di distribuzione.

Per visualizzare la ricetta nell'archivio, passa a appsetup.rb. L'esempio seguente mostra il codice di `appsetup.rb`.

Ad esempio`dbsetup.rb`, esegue `appsetup.rb` iterazioni sulle app nel `deploy` nodo, semplicemente simplephpapp, di nuovo. Esegue un blocco di codice con una risorsa `script` e una risorsa `template`.

La `script` risorsa installa Composer, un gestore di dipendenze per applicazioni PHP. Inoltre esegue il comando `install` di Composer per installare le dipendenze per l'applicazione di esempio nella directory principale dell'app.

La risorsa `template` genera `db-connect.php` e lo inserisce in `/srv/www/simplephpapp/current`. Tieni presente quanto segue:

- La ricetta utilizza un'istruzione condizionale per specificare il proprietario del file, che dipende dal sistema operativo dell'istanza.

- La direttiva `only_if` indica a Chef per generare il modello solo se la directory specificata esiste.

Una risorsa `template` opera su un modello che ha essenzialmente lo stesso contenuto e la stessa struttura del file associato, ma include segnaposto per i diversi valori di dati. Il parametro `source` specifica il modello, `db-connect.php.erb`, che si trova nella directory `templates/default` del libro di ricette phpapp e contiene quanto segue:

Quando Chef elabora il modello, sostituisce i segnaposto `<%= =>` con i valori delle variabili corrispondenti nella risorsa modello, che sono ricavati a loro volta dagli attributi di distribuzione. Il file generato sarà quindi:

Passaggio 3.3: Aggiungere i ricettari personalizzati a MyStack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Proprio come le app, puoi archiviare i libri di ricette personalizzati in un archivio. Ogni stack può avere un archivio contenente un set di libri di ricette personalizzati. Quindi indichi a AWS OpsWorks Stacks di installare i tuoi libri di cucina personalizzati sulle istanze dello stack.

1. Fare clic su Stack nel riquadro di navigazione per visualizzare la pagina dello stack corrente.
2. Fare clic su Stack Settings (Impostazioni stack), quindi su Edit (Modifica).
3. Modificare la configurazione dello stack nel seguente modo:
   - Usa libri di cucina Chef personalizzati — Sì
   - Tipo di repository: Git
   - URL del repository — **`git://github.com/amazonwebservices/opsworks-example-cookbooks.git`**
4. Fare clic su Save (Salva) per aggiornare la configurazione dello stack.



AWS OpsWorks Stacks installa quindi il contenuto del tuo repository di libri di cucina su tutte le istanze dello stack. Se crei nuove istanze, Stacks installa automaticamente il repository di libri di cucina AWS OpsWorks .

> **Note**
>
> Se devi aggiornare uno dei tuoi libri di cucina o aggiungere nuovi libri di cucina al repository, puoi farlo senza toccare le impostazioni dello stack. AWS OpsWorks Stacks installerà automaticamente i libri di cucina aggiornati su tutte le nuove istanze. Tuttavia, AWS OpsWorks Stacks non installa automaticamente i libri di cucina aggiornati sulle istanze online dello stack. È necessario indicare esplicitamente a AWS OpsWorks Stacks di aggiornare i libri di cucina eseguendo il comando stack. `Update Cookbooks` Per ulteriori informazioni, consulta Esecuzione dei comandi dello stack.

Fase 3.4. Esecuzione delle ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver creato il libro di ricette personalizzato, devi eseguire le ricette sulle istanze appropriate. Puoi eseguirle manualmente. Tuttavia, in genere le ricette devono essere eseguite in punti prevedibili del ciclo di vita di un'istanza, ad esempio dopo l'avvio dell'istanza o durante la distribuzione di un'app.

Questa sezione descrive un approccio molto più semplice: fai in modo che AWS OpsWorks Stacks li esegua automaticamente per te al momento opportuno.

AWS OpsWorks Stacks supporta una serie di eventi del ciclo di vita che semplificano l'esecuzione delle ricette. Ad esempio, l'evento Setup si verifica al termine dell'avvio di un'istanza, mentre l'evento Deploy si verifica durante la distribuzione di un'app. Ogni livello ha un set di ricette predefinite associate a ogni evento del ciclo di vita. Quando si verifica un evento del ciclo di vita su un'istanza, l'agente esegue le ricette associate per ogni livello dell'istanza. Per fare in modo che AWS OpsWorks Stacks esegua automaticamente una ricetta personalizzata, aggiungila all'evento del ciclo di vita appropriato sul livello appropriato e l'agente eseguirà la ricetta al termine delle ricette integrate.

Per questo esempio, è necessario eseguire due ricette, `dbsetup.rb` sull'istanza My SQLinstance e `appsetup.rb` sull'istanza PHP App Server.

> **Note**
>
> È possibile specificare le ricette sulla console utilizzando il *`recipe_name`* formato*`cookbook_name`*::, dove *`recipe_name`* non include l'estensione.rb. Ad esempio, puoi fare riferimento a `dbsetup.rb` utilizzando **`phpapp::dbsetup`**.

Per assegnare ricette personalizzate agli eventi del ciclo di vita

1. Nella pagina Livelli, per MySQL, fai clic su Ricette, quindi su Modifica.
2. Nella sezione Custom Chef recipes (Ricette di Chef personalizzate) immettere **`phpapp::dbsetup`** per Deploy (Distribuisci).



3. Fare clic sull'icona + per assegnare la ricetta all'evento, quindi fare clic su Save (Salve) per salvare la nuova configurazione del livello.
4. Tornate alla pagina Layers e ripetete la procedura da assegnare all'evento **`phpapp::appsetup`** Deploy del livello PHP App Server.

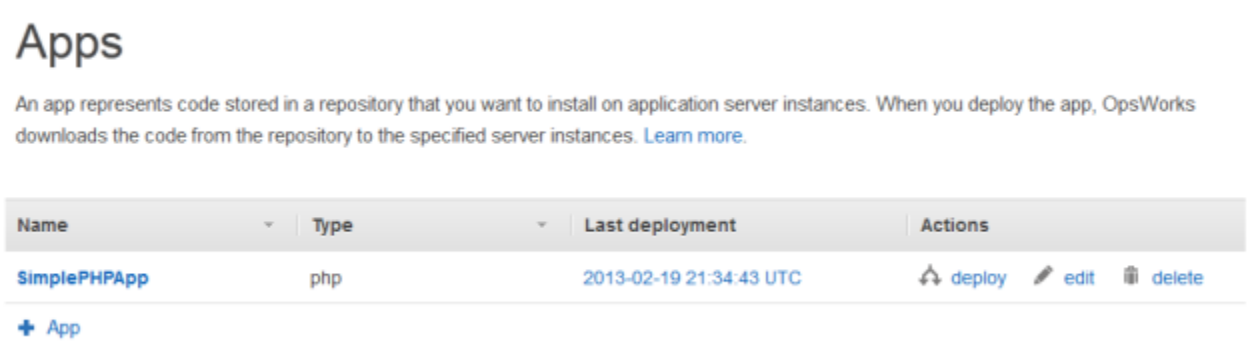
Fase 3.5: Deploy Simple, versione 2 PHPApp

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

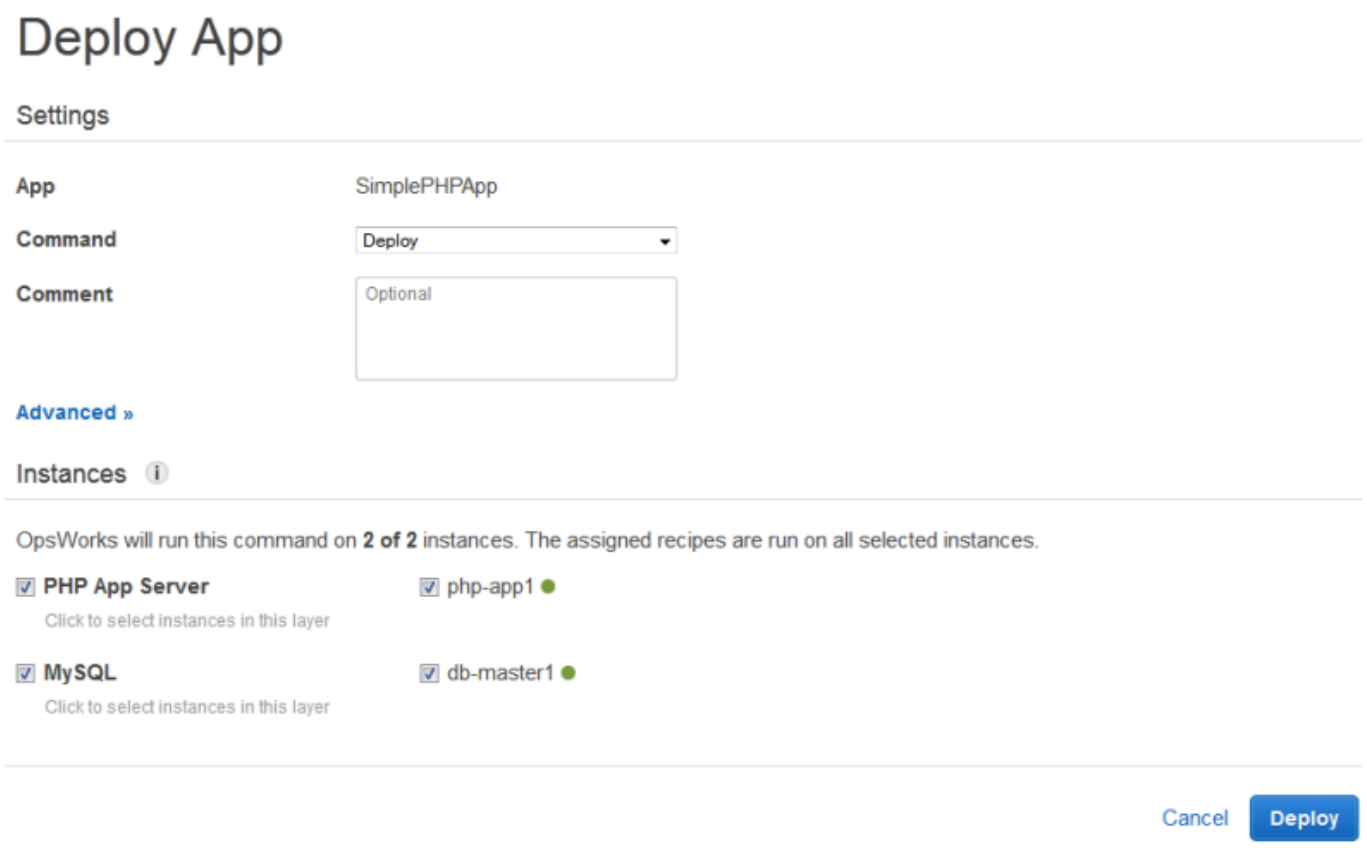
Il passaggio finale consiste nell'implementazione della nuova versione di Simple. PHPApp

Per distribuire Simple PHPApp

1. Nella pagina App, fai clic su Distribuisci nelle Azioni dell'PHPAppapp Simple.



2. Accettare le impostazioni predefinite e fare clic su Deploy (Distribuisci).



Quando fai clic su Deploy (Distribuisci) nella pagina Deploy App (Distribuisci applicazione), viene attivato un evento Deploy (Distribuzione) del ciclo di vita, che indica agli agenti di eseguire le rispettive ricette di distribuzione. Per impostazione predefinita, l'evento viene attivato su tutte le istanze dello stack. Le ricette Deploy integrate distribuiscono l'app solo nelle istanze appropriate per il tipo di app, in questo caso nelle istanze di PHP App Server. Tuttavia, spesso risulta utile attivare l'evento Deploy (Distribuzione) su altre istanze per consentire loro di rispondere alla distribuzione dell'app. In questo caso, vuoi anche attivare Deploy sull'istanza MySQL per configurare il database.

Tieni presente quanto segue:

- L'agente sull'istanza PHP App Server esegue la ricetta integrata del layer, seguita da `appsetup.rb`, che configura la connessione al database dell'app.
- L'agente sull'istanza MySQL non installa nulla, ma `dbsetup.rb` viene eseguito per creare la tabella degli urler.

Al termine della distribuzione, il campo Status (Stato) visualizzerà successful (operazione riuscita) nella pagina Deployment (Distribuzione).

Passaggio 3.6: Run Simple PHPApp

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo che lo stato di implementazione è stato modificato correttamente, puoi eseguire la nuova PHPApp versione Simple, come segue.

Per eseguire Simple PHPApp

1. Nella pagina Instances (Istanze) fare clic sull'indirizzo IP pubblico nella riga php-app1.

   Nel browser in uso dovrebbe venire visualizzata la seguente pagina.



2. Fare clic su Share Your Thought (Condividi i tuoi commenti) e digitare il testo desiderato, ad esempio **`Hello world!`** nel campo Your Thought (Commento) e immettere il proprio nome nel campo Your Name (Nome). Fare quindi clic su Submit Your Thought (Invia i tuoi commenti) per aggiungere il messaggio al database.

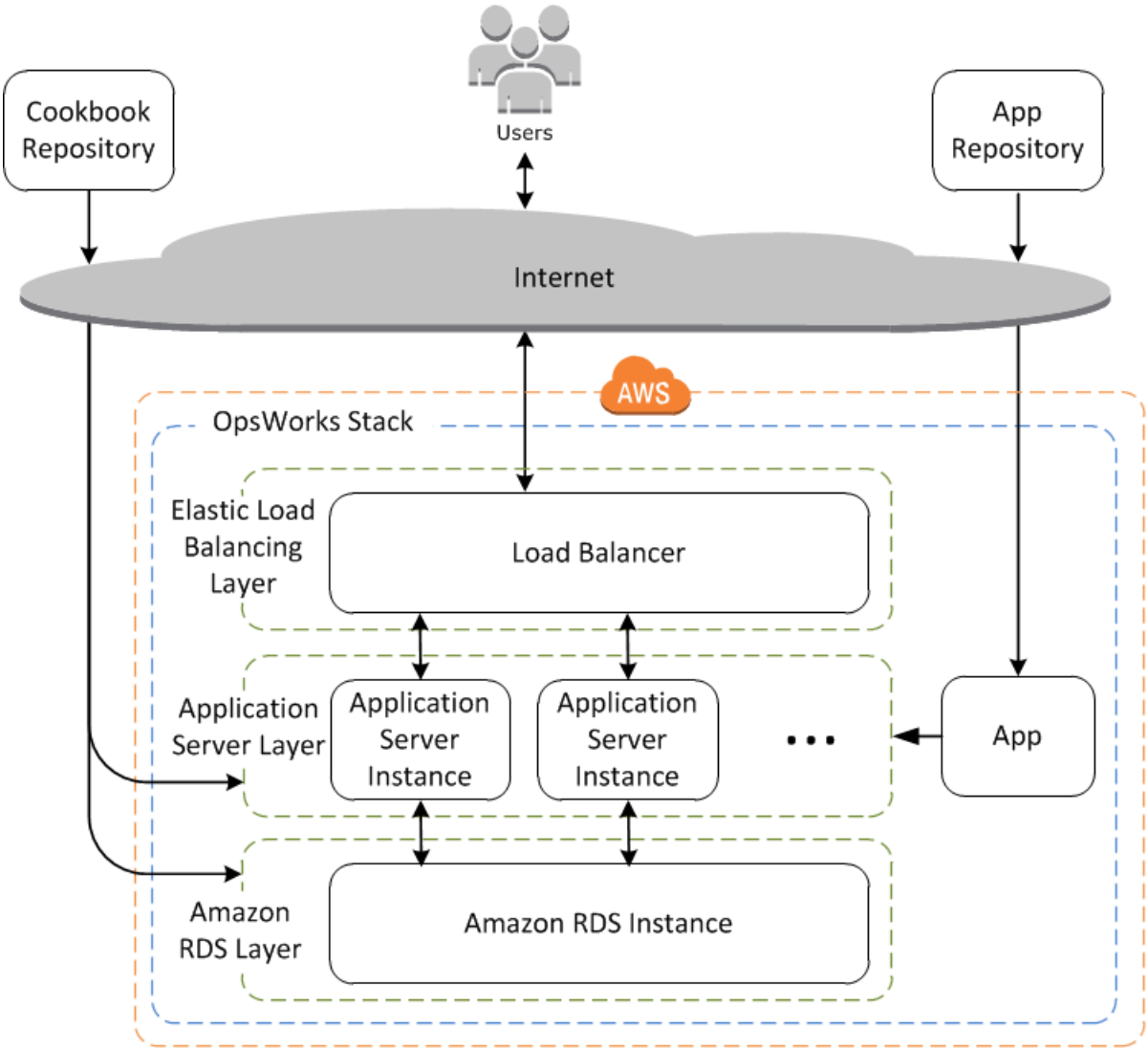


3. Fare clic su Go Back (Torna indietro) per visualizzare tutti i messaggi nel database.

Fase 4: Scalare Out MyStack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

MyStack attualmente dispone di un solo server di applicazioni. Uno stack di produzione probabilmente avrà più bisogno di più server di applicazioni per gestire il traffico in entrata e un sistema di bilanciamento del carico per distribuire il traffico in entrata in modo uniforme nei server di applicazioni. L'architettura sarà simile all'esempio seguente:



AWS OpsWorks Stacks semplifica la scalabilità orizzontale degli stack. Questa sezione descrive le basi su come scalare uno stack aggiungendo una seconda istanza di PHP App Server 24 ore su 24, 7 giorni su 7 MyStack e inserendo entrambe le istanze dietro un load balancer Elastic Load Balancing. Puoi facilmente estendere la procedura per aggiungere un numero arbitrario di istanze 24 ore su 24, 7 giorni su 7, oppure puoi utilizzare istanze basate sul tempo o sul carico per far sì che Stacks ridimensioni automaticamente lo stack. AWS OpsWorks Per ulteriori informazioni, consulta Gestione del carico con istanze basate sul tempo e sul carico.

Fase 4.1: aggiungere un sistema di bilanciamento del carico

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Elastic Load Balancing è un servizio AWS che distribuisce automaticamente il traffico delle applicazioni in entrata su più istanze Amazon. EC2 Oltre a distribuire il traffico, Elastic Load Balancing esegue le seguenti operazioni:

- Rileva istanze Amazon EC2 non integre.

  Re-instrada il traffico verso le istanze integre rimanenti, finché quelle non integre non vengono ripristinate.
- Ridimensiona automaticamente la capacità di gestione delle richieste in risposta al traffico in entrata

> **Note**
>
> Un sistema di bilanciamento del carico è in grado di elaborare due funzioni. La prima è certamente quella di equalizzare il carico di lavoro dei server dell'applicazione. Inoltre, molti siti preferiscono isolare i server di applicazioni e i database da accesso diretto degli utenti. Con AWS OpsWorks Stacks, puoi farlo eseguendo lo stack in un cloud privato virtuale (VPC) con una sottorete pubblica e privata, come segue.
>
> - Mettere i server di applicazioni e i database nella sottorete privata, dove altre istanze in VPC, ma non gli utenti, possono accedervi.
> - Traffico di utente diretto a un sistema di bilanciamento del carico nella sottorete pubblica, che inoltra quindi il traffico verso i server di applicazioni nella sottorete privata e restituisce risposte agli utenti.
>
> Per ulteriori informazioni, consulta Esecuzione di uno stack in un VPC. Per un AWS CloudFormation modello che estenda l'esempio di questa procedura dettagliata per l'esecuzione in un VPC, scarica il file. `OpsWorksVPCtemplates.zip`

Sebbene Elastic Load Balancing venga spesso definito un livello, funziona in modo leggermente diverso rispetto agli altri livelli incorporati. Invece di creare un layer e aggiungervi istanze, create un

load balancer Elastic Load Balancing utilizzando la console EC2 Amazon e poi collegatelo a uno dei livelli esistenti, di solito un livello di application server. AWS OpsWorks Stacks registra quindi le istanze esistenti del layer con il servizio e aggiunge automaticamente tutte le nuove istanze. La procedura seguente descrive come aggiungere un load balancer al MyStack livello PHP App Server.

> **Note**
>
> AWS OpsWorks Stacks non supporta Application Load Balancer. Puoi usare Classic Load Balancer solo con AWS OpsWorks Stacks.

Per collegare un sistema di bilanciamento del carico al livello PHP App Server

1. Usa la EC2 console Amazon per creare un nuovo sistema di bilanciamento del carico per MyStack. I dettagli dipendono dal fatto che il tuo account supporti o meno la EC2 versione Classic. Per ulteriori informazioni, consulta Nozioni di base su Elastic Load Balancing. Quando si esegue la procedura guidata Create Load Balancer (Crea sistema di bilanciamento del carico), configurare il sistema di bilanciamento del carico come segue:

   Define Load Balancer (Definisci sistema di bilanciamento del carico)

   Assegna al load balancer un nome facilmente riconoscibile, come PHP-LB, per facilitarne l'individuazione nella console Stacks. AWS OpsWorks Selezionare Continue (Continua) per accettare le impostazioni predefinite per le impostazioni restanti.

   Se si seleziona un VPC con una o più sottoreti dal menu Create LB Inside (Crea sistema di bilanciamento del carico all'interno), è necessario selezionare una sottorete per ogni zona di disponibilità in cui si desidera che il traffico venga instradato dal sistema di bilanciamento del carico.

   Assign Security Groups (Assegna i gruppi di sicurezza)

   Se il tuo account supporta una VPC predefinita, la procedura guidata visualizza questa pagina per determinare il gruppo di sicurezza del sistema di bilanciamento del carico. Non visualizza questa pagina per Classic. EC2

   Per questo scenario, selezionare default VPC security group (gruppo di sicurezza VPC predefinito).

Configure Security Settings (Configura impostazioni di sicurezza)

Se si seleziona HTTPS come Load Balancer Protocol (Protocollo sistema di bilanciamento del carico) nella pagina Define Load Balancer (Definisci sistema di bilanciamento del carico) configurare il certificato, cipher e le impostazioni del protocollo SSL in questa pagina. Per questo scenario, accettare le impostazioni predefinite e selezionare Configure Health Check (Configura controllo dello stato).

Configure Health Check (Configura controllo dello stato)

Impostare il percorso ping su **/** e accettare le impostazioni predefinite per le impostazioni restanti.

Aggiungi EC2 istanze

Scegli Continua; AWS OpsWorks Stacks registra automaticamente le istanze con il sistema di bilanciamento del carico.

Aggiungi tag

Aggiungere tag per aiutare a trovare . Ogni tag è una coppia chiave e valore; ad esempio, è possibile specificare **`Description`** come chiave e **`Test LB`** come valore ai fini di questo scenario.

Review (Revisione)

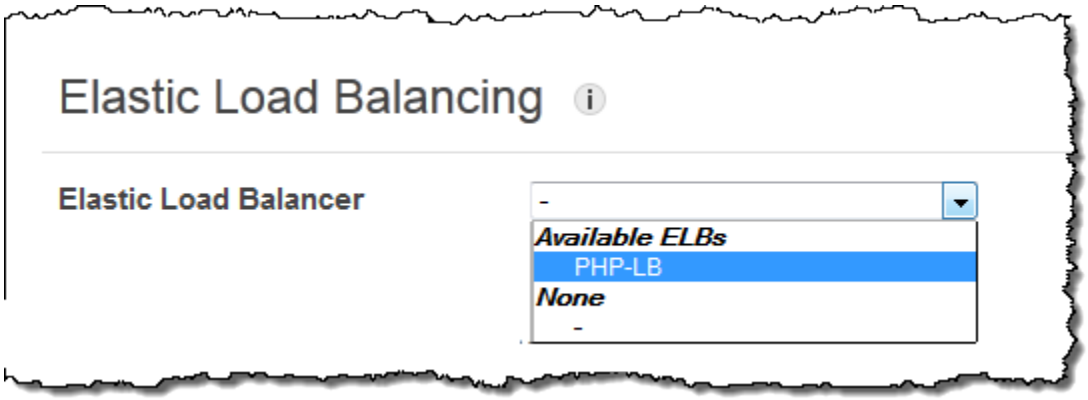
Esaminare le proprie scelte, selezionare Create (Crea) e in seguito Close (Chiudi), che avvia il sistema di bilanciamento del carico.

2. Se l'account supporta il VPC predefinito, dopo l'avvio del sistema di bilanciamento del carico, assicurarsi che il gruppo di sicurezza abbia regole di ingresso appropriate. La regola predefinita non accetta alcun traffico in ingresso.

   1. Scegli Security Groups nel pannello di EC2 navigazione di Amazon.
   2. Selezionare default VPC security group (gruppo di sicurezza VPC predefinito)
   3. Selezionare Edit (Modifica) nella scheda Inbound (In entrata).
   4. Per questo scenario, impostare Source (Origine) su Anywhere (Ovunque) per indicare al sistema di bilanciamento del carico di accettare il traffico in entrata da qualsiasi indirizzo IP.

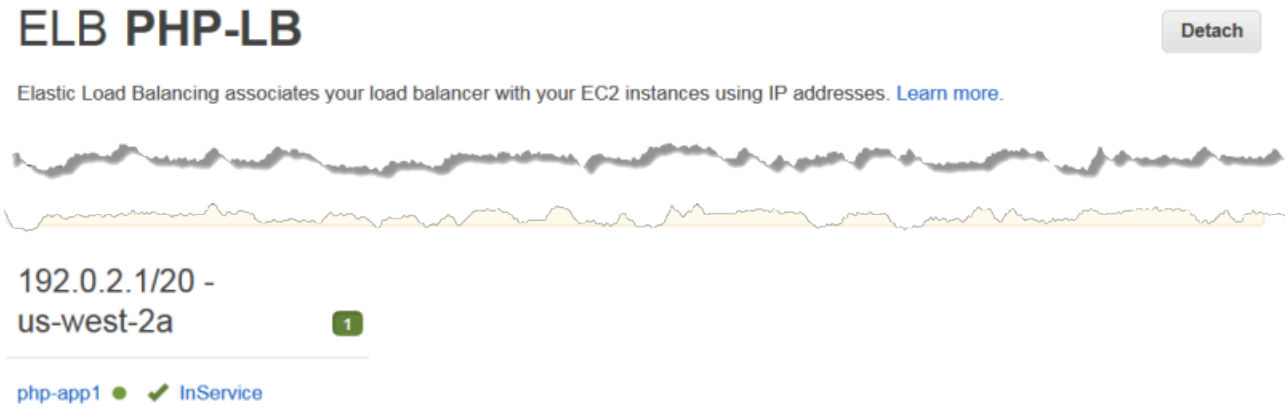
3. Torna alla console AWS OpsWorks Stacks. Nella pagina Layers (Livelli), selezionare il collegamento Network (Rete) del livello e in seguito Edit (Modifica).

4. In Elastic Load Balancing, selezionare il sistema di bilanciamento del carico creato nella Fase 1 e selezionare Save (Salva).



   Dopo aver collegato il load balancer al layer, AWS OpsWorks Stacks registra automaticamente le istanze correnti del layer e ne aggiunge di nuove non appena sono online.

5. Nella pagina Layers (Livelli), fare clic sul nome del sistema di bilanciamento del carico per aprire la pagina dei dettagli. Quando la registrazione è completa e l'istanza supera un controllo di integrità, AWS OpsWorks Stacks mostra un segno di spunta verde accanto all'istanza nella pagina del bilanciamento del carico.

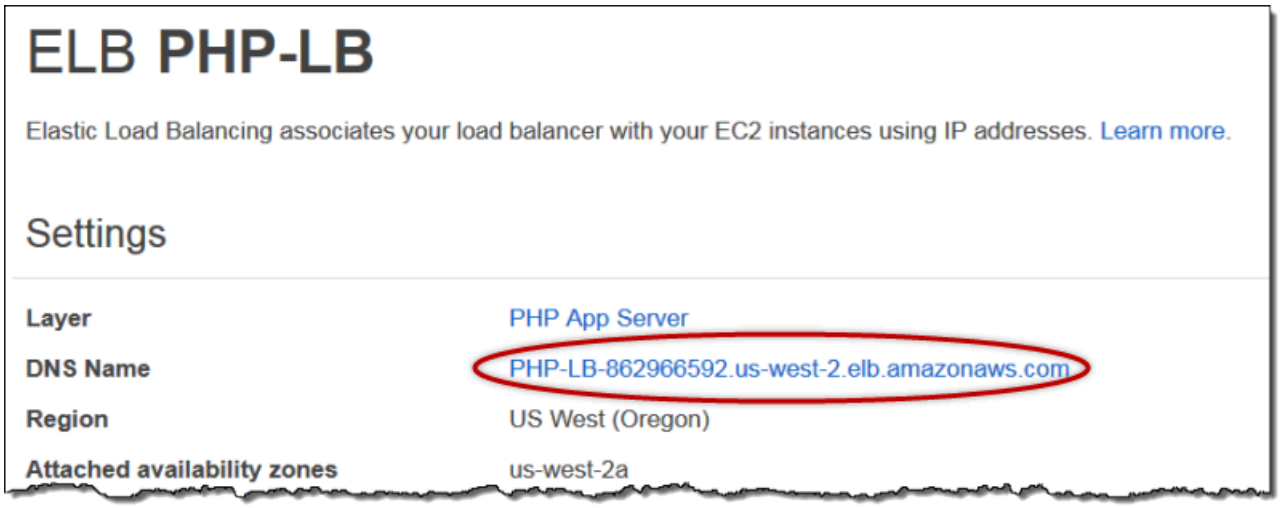


Ora puoi eseguire Simple PHPApp inviando una richiesta al sistema di bilanciamento del carico.

Per eseguire Simple PHPApp tramite il sistema di bilanciamento del carico

1. Aprire di nuovo la pagina dei dettagli del sistema di bilanciamento del carico, se non è già aperta.
2. Nella pagina delle proprietà, verifica lo stato del controllo dello stato dell'istanza e fai clic sul nome DNS del load balancer per eseguire Simple. PHPApp Il load balancer inoltra la richiesta all'istanza di PHP App Server e restituisce la risposta, che dovrebbe essere esattamente la

stessa della risposta che si ottiene quando si fa clic sull'indirizzo IP pubblico dell'istanza di PHP App Server.



> **Note**
>
> AWS OpsWorks Stacks supporta anche il HAProxy load balancer, che potrebbe avere dei vantaggi per alcune applicazioni. Per ulteriori informazioni, consulta HAProxy AWS OpsWorks Stacks Layer.

Passaggio 4.2: Aggiungere istanze PHP App Server

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ora che il load balancer è a posto, puoi scalare lo stack aggiungendo altre istanze al livello PHP App Server. Dal proprio punto di vista, l'operazione è fluida. Ogni volta che una nuova istanza di PHP App Server è online, AWS OpsWorks Stacks la registra automaticamente con il sistema di bilanciamento del carico e implementa SimplePHPApp, in modo che il server possa iniziare immediatamente a gestire il traffico in entrata. Per brevità, questo argomento mostra come aggiungere un'ulteriore

istanza di PHP App Server, ma puoi usare lo stesso approccio per aggiungerne quante ne hai bisogno.

Per aggiungere un'altra istanza al livello PHP App Server

1. Nella pagina Istanze, fai clic su + Instance in PHP App Server.
2. Accettare le impostazioni predefinite e fare clic su Add Instance (Aggiungi istanza).
3. Fare clic su start (avvia) per avviare l'istanza.

Fase 4.3: Monitoraggio MyStack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

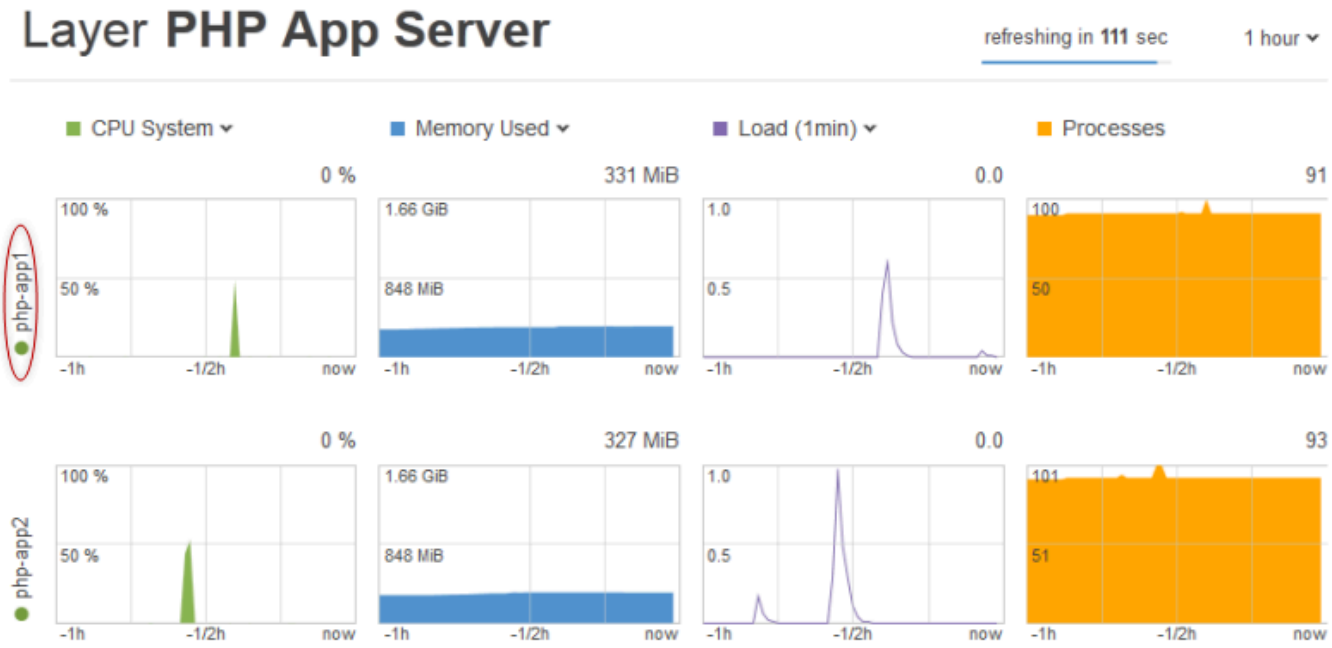
AWS OpsWorks Stacks utilizza Amazon CloudWatch per fornire i parametri per uno stack e li riassume per comodità nella pagina Monitoraggio. È possibile visualizzare i parametri per l'intero stack, un livello specificato o una determinata istanza.

Per monitorare MyStack

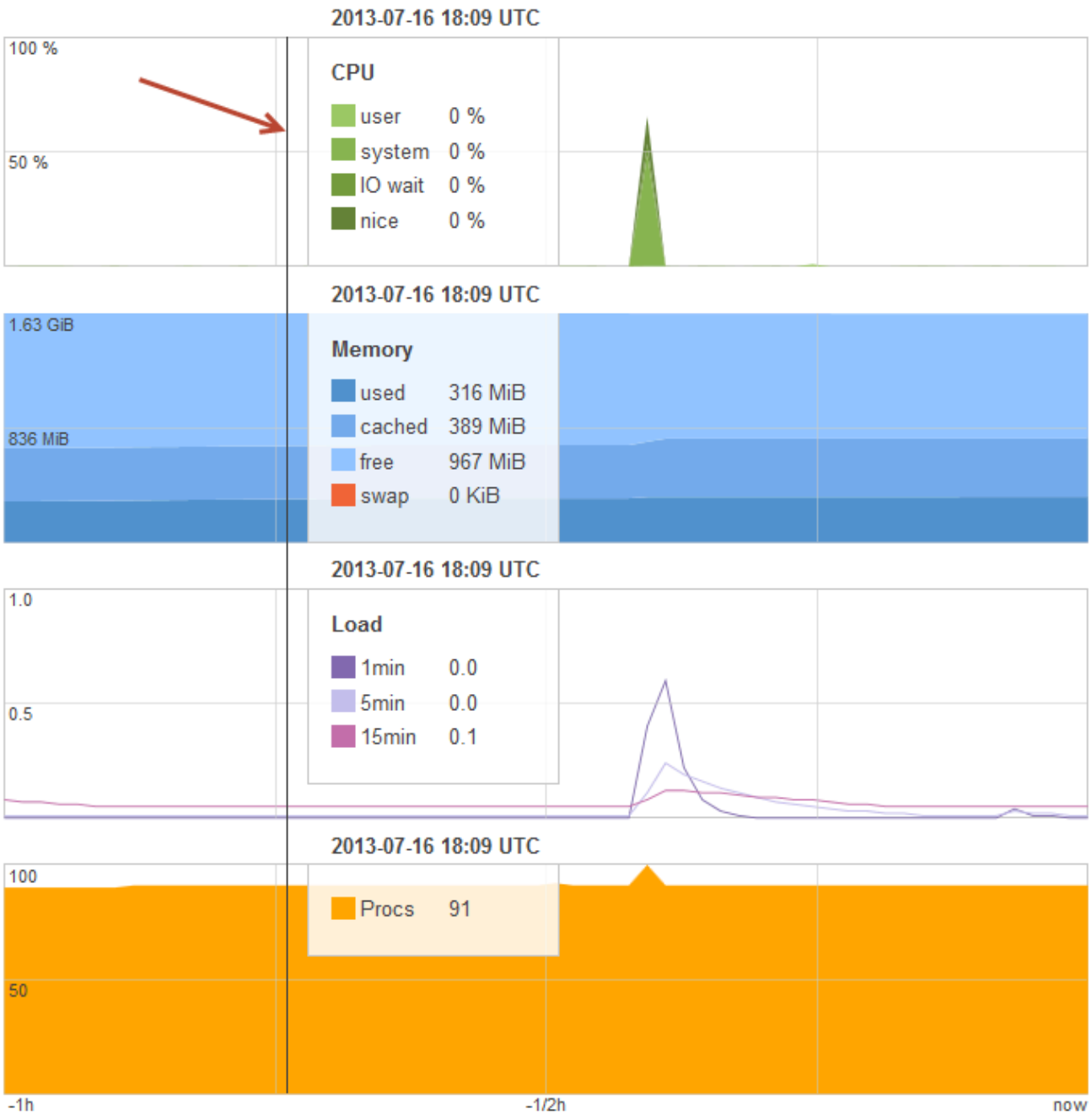
1. Nel riquadro di navigazione, fare clic su Monitoring (Monitoraggio), che mostra un set di grafici con parametri di media per ogni livello. È possibile utilizzare i menu per CPU System (Sistema CPU), Memory Used (Memoria utilizzata) e Load (Carico) per visualizzare i parametri correlati differenti.



2. Fare clic su PHP App Server per visualizzare i parametri per ciascuna delle istanze del livello.



3. Fare clic su php-app1 per visualizzare i parametri per quell'istanza. È possibile visualizzare i parametri per un determinato momento spostando il cursore.


Instance php-app1
refreshing in
2013-07-16 18:09 UTC
100 %
50 %
CPU
user 0 %
system 0 %
IO wait 0 %
nice 0 %
2013-07-16 18:09 UTC
1.63 GiB
836 MiB
Memory
used 316 MiB
cached 389 MiB
free 967 MiB
swap 0 KiB
2013-07-16 18:09 UTC
1.0
0.5
Load
1min 0.0
5min 0.0
15min 0.1
2013-07-16 18:09 UTC
100
50
Procs 91
-1h
-1/2h
now

> **Note**
>
> AWS OpsWorks Stacks supporta anche il server di monitoraggio Ganglia, che potrebbe avere vantaggi per alcune applicazioni. Per ulteriori informazioni, consulta Ganglia Layer.

Fase 5: Eliminare MyStack

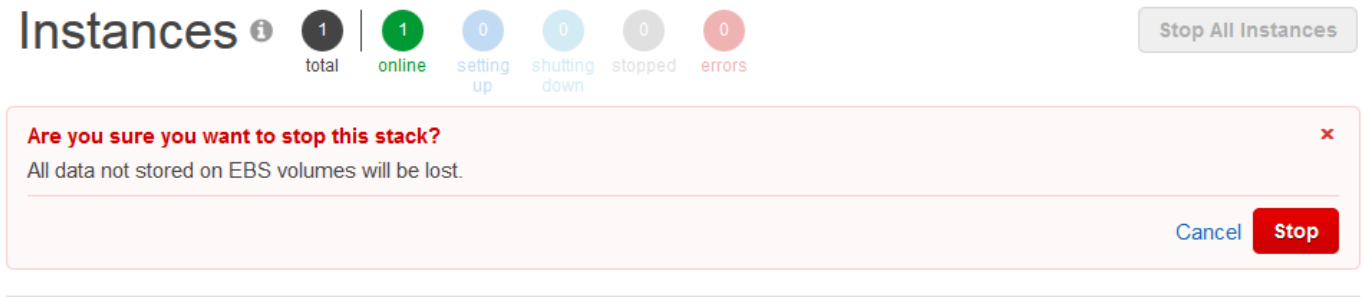
> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Non appena inizi a utilizzare risorse AWS come le EC2 istanze Amazon, ti verranno addebitati i costi in base all'utilizzo. Al termine dell'utilizzo, è consigliabile arrestare le istanze in modo da non dover sostenere costi indesiderati. Se non hai più bisogno dello stack, puoi eliminarlo.

Per eliminare MyStack

1. Arrestare tutte le istanze

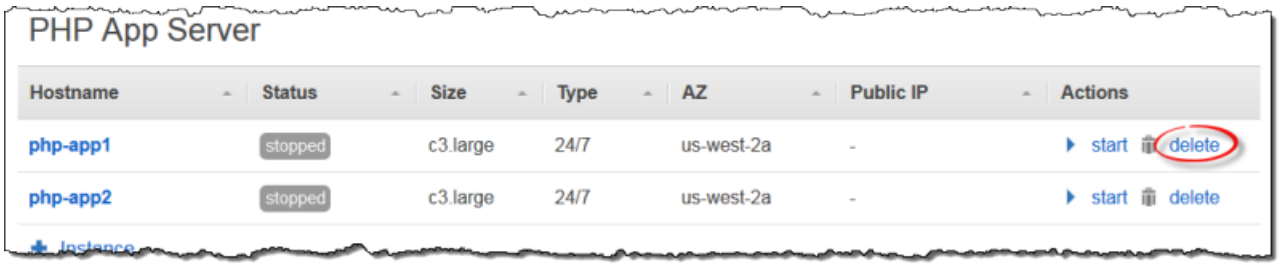
   Nella pagina Instances (Istanze), fare clic su Stop All Instances (Arresta tutte le istanze) e quindi su Stop (Arresta) quando viene richiesto di confermare l'operazione.



   Dopo aver fatto clic su Stop, AWS OpsWorks Stacks termina le EC2 istanze Amazon associate, ma non le risorse associate come gli indirizzi IP elastici o i volumi Amazon EBS.

2. Eliminare tutte le istanze

   L'arresto dell'istanza interrompe semplicemente le istanze Amazon EC2 associate. Dopo che lo stato delle istanze diventa 'stopped' (arrestato), devi eliminare ogni istanza. Nel livello PHP App Server (Server app PHP) fare clic su delete (eliminazione) nella colonna Actions (Operazioni) dell'istanza php-app1.

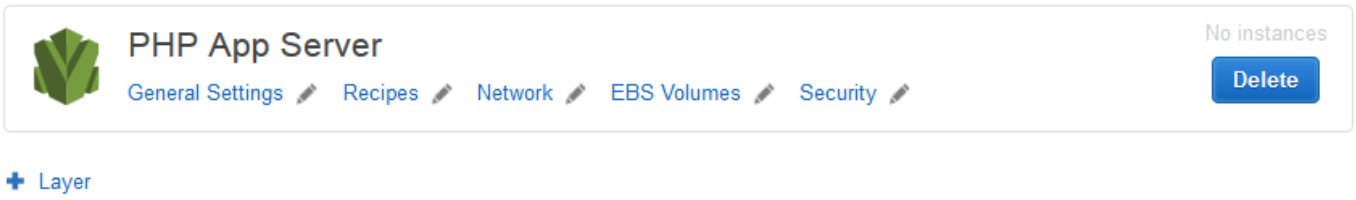


   AWS OpsWorks Stacks ti chiede quindi di confermare l'eliminazione e ti mostra tutte le risorse dipendenti. È possibile scegliere di conservare una o tutte queste risorse. Questo esempio non prevede risorse dipendenti, perciò fare clic su Delete (Elimina).

   Ripetete il processo per php-app2 e l'istanza MySQL db-master1. Nota che db-master1 ha un volume Amazon Elastic Block Store associato, selezionato per impostazione predefinita. Lasciarlo selezionata per eliminare il volume assieme all'istanza.

3. Eliminare i livelli.

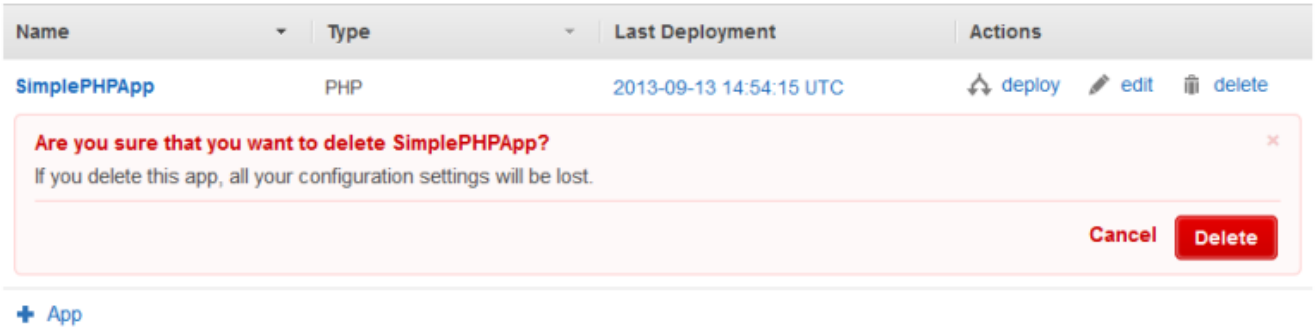
   Nella pagina Layers (Livelli), fare clic su Delete (Elimina) e quindi su Delete (Elimina) per confermare.



   Ripetere il processo per il livello MySQL.

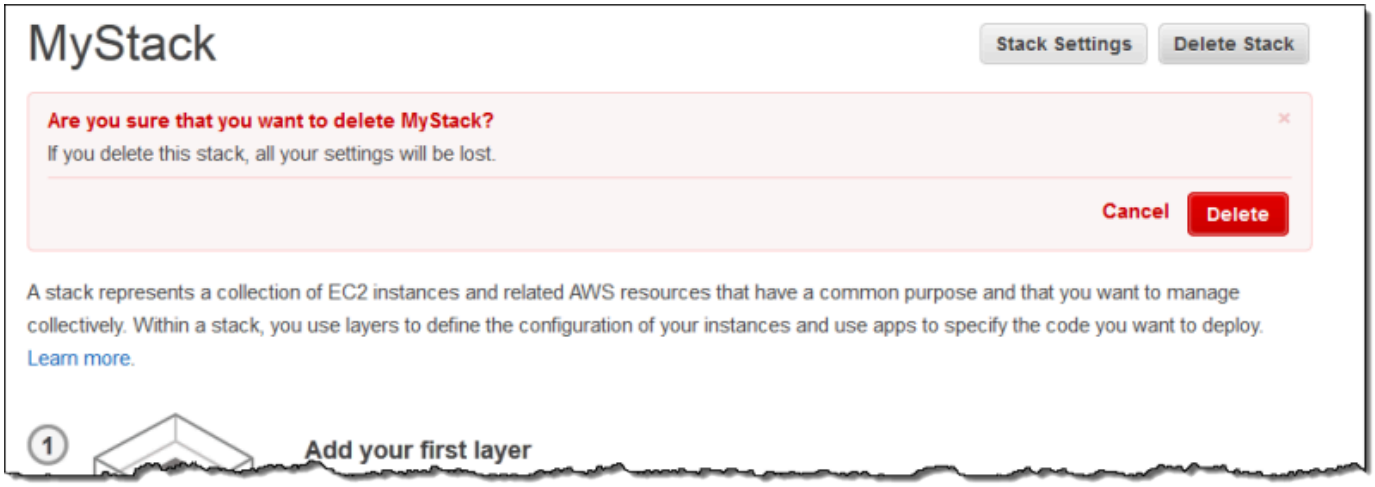
4. Eliminare l'app

   Nella pagina App, fai clic su Elimina nella colonna Azioni dell'PHPAppapp Simple, quindi fai clic su Elimina per confermare.



5. Elimina MyStack

   Nella pagina Stack, fare clic su Delete Stack (Elimina stack) e quindi su Delete (Elimina) per confermare.



Hai completato questa procedura guidata.

Creazione del primo stack Node.js

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo esempio illustra come creare uno stack Linux che supporta un server di applicazioni Node.js e come distribuire un'applicazione semplice. Lo stack è composto dai componenti seguenti:

- Un livello di App Server Node.js con due istanze
- Un sistema di bilanciamento del carico Elastic Load Balancing per distribuire il traffico verso le istanze del server delle applicazioni
- Un livello di servizio Amazon Relational Database Service (Amazon RDS) che fornisce un database di backend

Argomenti

- Prerequisiti
- Implementazione dell'applicazione
- Creazione del server di database e di un sistema di bilanciamento del carico
- Creazione dello stack
- Distribuzione dell'applicazione
- Come procedere

Prerequisiti

Per questa procedura guidata sono validi i requisiti riportati di seguito:

- Hai un account AWS e una conoscenza di base di come usare AWS OpsWorks Stacks.

  Se non conosci AWS OpsWorks Stacks o AWS, impara le nozioni di base completando il tutorial introduttivo in. Nozioni di base sugli stack Linux Chef 11
- Disponi di conoscenze di base relative alle procedure necessarie per implementare un'applicazione Node.js.

  Se sei un utente principiante di Node.js, per approfondire le nozioni di base, completa un tutorial introduttivo, ad esempio quello relativo ai nodi.
- Hai già creato almeno uno stack nella regione AWS che prevedi di utilizzare per questo esempio.

  Quando crei il primo stack in una regione, AWS OpsWorks Stacks crea un gruppo di sicurezza Amazon Elastic Compute Cloud (Amazon EC2) per ogni tipo di livello. Questi gruppi di sicurezza sono necessari per creare l'istanza del database Amazon RDS (DB). Se non conosci AWS

OpsWorks Stacks, ti consigliamo di utilizzare per questo esempio la stessa regione che hai usato quando hai seguito il tutorial. Nozioni di base sugli stack Linux Chef 11 Se vuoi utilizzare una nuova regione, crea un nuovo stack nella regione. Non è necessario che lo stack includa livelli o istanze. Non appena crei lo stack, AWS OpsWorks Stacks aggiunge automaticamente un set di gruppi di sicurezza alla regione.

- Dovrai creare lo stack in un VPC predefinito.

  Puoi usare EC2 -Classic per questa procedura dettagliata, ma alcuni dettagli differiranno leggermente. Ad esempio, con EC2 -Classic, si specifica la zona di disponibilità (AZ) di un'istanza anziché la relativa sottorete.

- Il tuo utente IAM dispone delle autorizzazioni di accesso completo per Stacks. AWS OpsWorks

  Per motivi di sicurezza, ti consigliamo di non utilizzare le credenziali root del tuo account per questa procedura guidata. Invece, crea un utente con autorizzazioni di accesso completo a AWS OpsWorks Stacks e usa tali credenziali con Stacks. AWS OpsWorks Per ulteriori informazioni, consulta Creazione di un utente amministratore .

Implementazione dell'applicazione

Questa procedura dettagliata utilizza una semplice applicazione Express che si connette all'istanza DB di Amazon RDS ed elenca i database dell'istanza.

Per implementare l'applicazione, crea una directory denominata `nodedb` in una posizione comoda sulla workstation e aggiungi i seguenti tre file.

Argomenti

- Descrittore del pacchetto
- File di layout
- File di codice

Descrittore del pacchetto

Per creare il descrittore del pacchetto dell'applicazione, aggiungi un file denominato `package.json` on i seguenti contenuti nella directory `nodedb`. `package.json` è obbligatorio per le applicazioni Express e deve trovarsi nella directory radice dell'applicazione.

```
{
  "name": "Nodejs-DB",
  "description": "Node.js example application",
  "version": "0.0.1",
  "dependencies": {
    "express": "*",
    "ejs": "*",
    "mysql": "*"
  }
}
```

Questo descrittore `package.json` è un esempio minimo. Definisce gli attributi obbligatori `name` e `version` ed elenca i pacchetti dipendenti:

- `express` fa riferimento al pacchetto Express.
- `ejs` fa riferimento al pacchetto EJS, utilizzato dall'applicazione per inserire testo in un file di layout HTML.
- `mysql` fa riferimento al pacchetto node-mysql, utilizzato dall'applicazione per connettersi all'istanza RDS.

Per ulteriori informazioni sui file dei descrittori dei pacchetti, consulta package.json.

File di layout

Per creare il file di layout dell'applicazione, aggiungi una directory `views` alla directory `nodedb`, quindi aggiungi un file a `views` denominato `index.html` con i seguenti contenuti:

```
<!DOCTYPE html>
<html>
<head>
  <title>AWS Opsworks Node.js Example</title>
</head>
<body>
  <h1>AWS OpsWorks Node.js Example</h1>
    <p>Amazon RDS Endpoint: <i><%= hostname %></i></p>
    <p>User: <i><%= username %></i></p>
    <p>Password: <i><%= password %></i></p>
    <p>Port: <i><%= port %></i></p>
    <p>Database: <i><%= database %></i></p>
```

```
    <p>Connection: <%= connectionerror %></p>
    <p>Databases: <%= databases %></p>
</body>
</html>
```

Per questo esempio, il file di layout è un semplice documento HTML che mostra alcuni dati di Amazon RDS. Ogni elemento `<%= ... =>` rappresenta il valore di una variabile definita nel file di codice dell'applicazione. Quest'ultimo file verrà creato in seguito.

File di codice

Per creare il file di codice dell'applicazione, aggiungi un file `server.js` alla directory `nodedb` con i seguenti contenuti.

> **Important**
>
> Con AWS OpsWorks Stacks, il file di codice principale di un'applicazione Node.js deve avere un nome `server.js` e trovarsi nella cartella principale dell'applicazione.

```
var express = require('express');
var mysql = require('mysql');
var dbconfig = require('opsworks'); //[1] Include database connection data
var app = express();
var outputString = "";

app.engine('html', require('ejs').renderFile);

//[2] Get database connection data
app.locals.hostname = dbconfig.db['host'];
app.locals.username = dbconfig.db['username'];
app.locals.password = dbconfig.db['password'];
app.locals.port = dbconfig.db['port'];
app.locals.database = dbconfig.db['database'];
app.locals.connectionerror = 'successful';
app.locals.databases = '';

//[3] Connect to the Amazon RDS instance
var connection = mysql.createConnection({
    host: dbconfig.db['host'],
```

```
    user: dbconfig.db['username'],
    password: dbconfig.db['password'],
    port: dbconfig.db['port'],
    database: dbconfig.db['database']
});

connection.connect(function(err)
{
    if (err) {
        app.locals.connectionerror = err.stack;
        return;
    }
});

// [4] Query the database
connection.query('SHOW DATABASES', function (err, results) {
    if (err) {
        app.locals.databases = err.stack;
    }

    if (results) {
        for (var i in results) {
            outputString = outputString + results[i].Database + ', ';
        }
        app.locals.databases = outputString.slice(0, outputString.length-2);
    }
});

connection.end();

app.get('/', function(req, res) {
    res.render('./index.html');
});

app.use(express.static('public'));

//[5] Listen for incoming requests
app.listen(process.env.PORT);
```

Nell'esempio vengono visualizzate le informazioni sulla connessione al database, viene eseguita una query sul server di database e vengono visualizzati i database del server. Puoi generalizzare in modo da interagire con il database in base alle esigenze. Le seguenti note si riferiscono ai commenti numerati nel codice precedente.

[1] Include database connection data (Inclusione dei dati di connessione al database)

Questa istruzione `require` include i dati di connessione al database. Come descritto più avanti, quando si collega un'istanza di database a un'app, AWS OpsWorks Stacks inserisce i dati di connessione in un file denominato`opsworks.js`, simile al seguente:

```
exports.db = {
  "host":"nodeexample.cdlqlk5uwd0k.us-west-2.rds.amazonaws.com",
  "database":"nodeexampledb",
  "port":3306,
  "username":"opsworksuser",
  "password":"your_pwd",
  "reconnect":true,
  "data_source_provider":"rds",
  "type":"mysql"}
```

`opsworks.js` si trova nella directory `shared/config` dell'applicazione, ovvero `/srv/www/`*`app_shortname`*`/shared/config`. Tuttavia, AWS OpsWorks Stacks inserisce un collegamento simbolico `opsworks.js` nella directory principale dell'applicazione, quindi puoi includere l'oggetto usando just. `require 'opsworks'`

[2] Get database connection data (Recupero dei dati di connessione al database)

Questo set di istruzioni mostra i dati di connessione di `opsworks.js` mediante l'assegnazione dei valori dell'oggetto `db` a un set di proprietà `app.locals`, ciascuna delle quali è mappata a uno degli elementi <%= ... %> nel file `index.html`. Nel documento visualizzato, gli elementi <%= ... %> sono sostituiti dai corrispondenti valori delle proprietà.

[3] Connect to the Amazon RDS instance (Connessione all'istanza Amazon RDS)

L'esempio utilizza `node-mysql` per accedere al database. Per connettersi al database, l'esempio crea un oggetto `connection` mediante l'invio dei dati di connessione a `createConnection` e quindi mediante l'esecuzione di una chiamata a `connection.connect` per stabilire la connessione.

[4] Query the database (Esecuzione di query sul database)

Dopo aver stabilito una connessione, l'esempio esegue una chiamata a `connection.query` per eseguire query sul database. Questo esempio effettua semplicemente l'esecuzione di query per i nomi dei database del server. `query` restituisce una matrice di oggetti `results`, uno per ogni database, con il nome del database assegnato alla proprietà `Database`. L'esempio concatena

i nomi e li assegna ad `app.locals.databases`, che visualizza l'elenco nella pagina HTML visualizzata.

Per questo esempio, ci sono cinque database, il `nodeexampledb` database che hai specificato quando hai creato l'istanza RDS e altri quattro che vengono creati automaticamente da Amazon RDS.

[5] Listen for incoming requests (Ascolto delle richieste in arrivo)

L'istruzione finale rimane in ascolto delle richieste in entrata su una porta specifica. Non devi specificare un valore di porta esplicito. Quando aggiungi l'app allo stack, specifichi se l'applicazione supporta le richieste HTTP o HTTPS. AWS OpsWorks Stacks imposta quindi la variabile di `PORT` ambiente su 80 (HTTP) o 443 (HTTPS) e puoi utilizzare quella variabile nell'applicazione.

È possibile ascoltare su altre porte, ma il gruppo di sicurezza integrato del livello di Node.js App Server, AWS- OpsWorks -NodeJS-App-Server, consente il traffico utente in entrata solo verso le porte 80, 443 e 22 (SSH). Per consentire il traffico utente in entrata verso altre porte, crea un gruppo di sicurezza con regole in entrata appropriate e assegnalo al livello Node.js App Server. Non modificare le regole in entrata modificando il gruppo di sicurezza predefinito. Ogni volta che crei uno stack, AWS OpsWorks Stacks sovrascrive i gruppi di sicurezza integrati con le impostazioni standard, quindi tutte le modifiche apportate andranno perse.

> **Note**
>
> Puoi associare variabili di ambiente personalizzate all'applicazione quando crei o aggiorni l'app. Puoi anche trasferire i dati all'applicazione utilizzando il formato JSON personalizzato e una ricetta personalizzata. Per ulteriori informazioni, consulta Passaggio di dati alle applicazioni.

Creazione del server di database e di un sistema di bilanciamento del carico

Questo esempio utilizza il server di database Amazon RDS e le istanze di bilanciamento del carico Elastic Load Balancing. Devi creare ogni istanza separatamente e quindi integrarla nello stack. Questa sezione descrive come creare nuove istanze di database e nuove istanze del sistema di bilanciamento del carico. Puoi anche utilizzare le istanze esistenti, ma ti consigliamo di fare riferimento alla procedura per verificare che tali istanze siano state configurate correttamente.

Di seguito viene descritto come creare un'istanza di database RDS con una configurazione minima sufficiente per questo esempio. Per ulteriori informazioni, consultare la Guida per l'utente di Amazon RDS.

Per creare un'istanza di database RDS

1. Apri la console.

   Apri la console Amazon RDS e imposta la regione su Stati Uniti occidentali (Oregon). Nel riquadro di navigazione, scegliere RDS Dashboard (Pannello di controllo RDS), quindi selezionare Launch DB Instance (Avvia istanza database).

2. Specificare il nome del motore del database.

   Scegliere MySQL Community Edition come motore di database.

3. Non abilitare l'implementazione Multi-AZ.

   Scegliere No, this instance... (No, questa istanza...) e quindi scegliere Next (Avanti). In questo esempio non è necessaria l'implementazione Multi-AZ.

4. Configurare le impostazioni di base.

   Nella pagina DB Instance Details (Dettagli istanza database), specificare le seguenti impostazioni:

   - DB Instance Class (Classe di istanza database): db.t2.micro
   - Multi-AZ deployment (Implementazione Multi-AZ): No
   - Allocated Storage (Storage allocato): **5** GB
   - DB Instance Identifier (Identificatore istanze DB): **`nodeexample`**
   - Master Username (Nome utente master): **`opsworksuser`**.
   - Master Password (Password master): password personalizzata

   Annotare l'identificatore dell'istanza, il nome utente e la password per un utilizzo successivo, accettare le impostazioni predefinite per le altre opzioni, quindi scegliere Next (Avanti).

5. Configurare le impostazioni avanzate.

   Nella pagina Configure Advanced Settings (Configura impostazioni avanzate), specificare le seguenti impostazioni:

- Database Name (Nome del database): **nodeexampledb**
- Gruppi di sicurezza DB: AWS - OpsWorks -DB-master-Server

> **Note**
>
> Il gruppo di sicurezza AWS- OpsWorks -DB-Master-Server consente solo alle istanze dello stack di accedere al database. Se si desidera accedere al database direttamente, collegare un gruppo di sicurezza aggiuntivo all'istanza di database RDS con le regole in entrata adeguate. Per ulteriori informazioni, consulta Gruppi di sicurezza Amazon RDS. È inoltre possibile controllare l'accesso inserendo l'istanza in un VPC. Per ulteriori informazioni, consulta Esecuzione di uno stack in un VPC.

Annotare il nome del database per un utilizzo successivo, accettare i valori predefiniti delle altre impostazioni, quindi scegliere Launch DB Instance (Avvia istanza database).

La procedura seguente descrive come creare un sistema di bilanciamento del carico Elastic Load Balancing per questo esempio. Per ulteriori informazioni, consulta la Guida per l'utente di Elastic Load Balancing.

Per creare il sistema di bilanciamento del carico

1. Apri la EC2 console Amazon.

   Apri la EC2 console Amazon e assicurati che la regione sia impostata su Stati Uniti occidentali (Oregon). Nel riquadro di navigazione sinistro, selezionare Load Balancers (Sistemi di bilanciamento del carico), quindi Create Load Balancer (Crea sistema di bilanciamento del carico).
2. Definire il sistema di bilanciamento del carico.

   Nella pagina Define Load Balancer (Definisci sistema di bilanciamento del carico), specificare le seguenti impostazioni.

   - Nome: **Node-LB**
   - Crea LB Inside: il mio VPC predefinito

Accettare le impostazioni predefinite per le altre opzioni e fare clic su Next (Avanti).

3. Assegnare i gruppi di sicurezza.

   Nella pagina Assign Security Groups (Assegna i gruppi di sicurezza), specificare i seguenti gruppi:

   - default VPC security group (gruppo di sicurezza VPC predefinito)
   - AWS- OpsWorks -NodeJS-App-Server

   Scegli Next (Successivo). Nella pagina Configure Security Settings (Configura impostazioni di sicurezza) scegliere Next (Avanti). Non è necessario disporre di un listener sicuro in questo esempio.

4. Configurare il controllo dello stato.

   Nella pagina Configure Health Check (Configura controllo dello stato), impostare Ping Path (Percorso ping) su **/** e accettare i valori predefiniti per le altre impostazioni. Scegli Next (Successivo). Nella pagina Aggiungi istanze, scegli Avanti. EC2 Nella pagina Aggiungi tag, scegli Rivedi e crea. AWS OpsWorks Stacks si occupa dell'aggiunta di EC2 istanze al load balancer e per questo esempio non avrai bisogno di tag.

5. Creare il sistema di bilanciamento del carico.

   Nella pagina Review (Rivedi), scegliere Create (Crea) per creare il sistema di bilanciamento del carico.

Creazione dello stack

A questo punto disponi di tutti i componenti necessari per creare lo stack.

Per creare lo stack

1. Accedi alla console Stacks. AWS OpsWorks

   Effettuare l'accesso alla console AWS OpsWorks Stacks e scegliere Add Stack (Aggiungi stack).

2. Creare lo stack.

   Per creare un nuovo stack, scegliere Chef 11 stack (Stack Chef 11), quindi specificare le seguenti impostazioni.

- – **`NodeStack`**
- Regione: Stati Uniti occidentali (Oregon)

  Puoi creare uno stack in qualsiasi regione AWS, ma per i tutorial consigliamo Stati Uniti occidentali (Oregon).

Selezionare Add Stack (Aggiungi stack). Per ulteriori informazioni sulle impostazioni di configurazione dello stack, consulta Creare un nuovo stack.

3. Aggiungi un livello di App Server Node.js con un sistema di bilanciamento del carico collegato.

   Nella NodeStackpagina, scegliete Aggiungi un layer, quindi specificate le seguenti impostazioni:

   - Tipo di livello: Node.js App Server
   - Elastic Load Balancer — Node-LB

   Accettare i valori predefiniti per le altre impostazioni, quindi scegliere Add Layer (Aggiungi livello).

4. Aggiungere istanze al livello e avviarle.

   Nel riquadro di navigazione, scegliere Instances (Istanze), quindi aggiungere due istanze al livello del server di applicazioni Rails, come descritto di seguito.

   1. In Node.js App Server, scegli Aggiungi istanza.

      Impostare Size (Dimensione) su t2.micro, accettare i valori predefiniti per le altre impostazioni, quindi scegliere Add Instance (Aggiungi istanza).

   2. Scegliere +Instance (+Istanza), quindi aggiungere una seconda istanza t2.micro al livello in una sottorete diversa.

      Ciò inserisce l'istanza in una zona di disponibilità separata.

   3. Scegliere Add instance (Aggiungi istanza).
   4. Per avviare entrambe le istanze, scegliere Start All Instances (Avvia tutte le istanze).

   È stato assegnato un sistema di bilanciamento del carico Elastic Load Balancing a questo layer. Quando un'istanza entra o esce dallo stato online, AWS OpsWorks Stacks registra o annulla automaticamente la registrazione dell'istanza con il sistema di bilanciamento del carico.

> **Note**
>
> Per uno stack di produzione, consigliamo di distribuire le istanze del server delle applicazioni su più istanze. AZs Se gli utenti non possono connettersi a una zona di disponibilità, il sistema di bilanciamento del carico instrada il traffico in entrata alle istanze nelle zone rimanenti e il sito continuerà a funzionare.

5. Registrare l'istanza di database RDS con lo stack.

   Nel riquadro di navigazione, scegliere Resources (Risorse) e registrare l'istanza di database RDS con lo stack, come segue.

   1. Scegliere la scheda RDS, quindi scegliere Show Unregistered RDS DB instances (Mostra istanze database RDS non registrate).
   2. Scegliere l'istanza nodeexampledb, quindi specificare le impostazioni seguenti:
      - Utente: il nome utente principale specificato al momento della creazione dell'istanza, per questo esempio. **`opsworksuser`**.
      - Password: la password principale specificata al momento della creazione dell'istanza.
   3. Scegli Register with Stack per aggiungere l'istanza DB RDS allo stack come livello di servizio Amazon RDS.

   > **Warning**
   >
   > AWS OpsWorks Stacks non convalida i valori User o Password, ma li passa semplicemente all'applicazione. Se questi valori non vengono immessi correttamente, l'applicazione non è in grado di connettersi al database.

   Per aggiungere l'istanza DB RDS allo stack come livello di servizio Amazon RDS, scegli Register with Stack.

Distribuzione dell'applicazione

Devi archiviare l'applicazione in un archivio remoto. Quando la distribuisci, AWS OpsWorks Stacks distribuisce il codice e i file correlati dal repository alle istanze del server delle applicazioni. Per comodità, questo esempio utilizza un archivio Amazon Simple Storage Service (Amazon S3) pubblico

come repository, ma puoi anche utilizzare diversi altri tipi di repository, tra cui Git e Subversion. Per ulteriori informazioni, consulta Origine dell'applicazione.

Per distribuire un'applicazione

1. Creare il pacchetto dell'applicazione in un file di archivio.

   Creare un archivio `.zip` della directory `nodedb` e delle relative sottodirectory denominando il pacchetto nodedb.zip. È inoltre possibile utilizzare altri tipi di file di archivio, tra cui gzip, bzip2 o tarball. Tieni presente che AWS OpsWorks Stacks non supporta archivi tar non compressi. Per ulteriori informazioni, consulta Origine dell'applicazione.

2. Carica il file di archivio su Amazon S3.

   Carica `nodedb.zip` in un bucket Amazon S3, rendi pubblico il file e copia l'URL del file per un uso successivo. Per ulteriori informazioni su come creare bucket e caricare file, consulta Nozioni di base su Amazon Simple Storage Service.

   > **Note**
   >
   > AWS OpsWorks Gli stack possono anche distribuire file privati da un bucket Amazon S3, ma per semplicità, questo esempio utilizza un file pubblico. Per ulteriori informazioni, consulta Origine dell'applicazione.

3. Crea un'app Stacks. AWS OpsWorks

   Torna alla console AWS OpsWorks Stacks, nel riquadro di navigazione, scegli App, quindi scegli Aggiungi un'app. Specificare le seguenti impostazioni:

   - Nome: `NodeDB`.

     Questa stringa è il nome visualizzato dell'applicazione. Per la maggior parte degli scopi, è necessario il nome breve dell'app, che AWS OpsWorks Stacks genera dal nome visualizzato trasformando tutti i caratteri in minuscolo e rimuovendo la punteggiatura. In questo esempio il nome è `nodedb`. Per verificare il nome breve di un'app, dopo aver creato l'app, scegliere l'app nella pagina Apps (App) per visualizzare la relativa pagina dei dettagli.
   - Tipo – `Node.js`.
   - Data source type (Tipo di origine dati) – `RDS`.

- Istanza di database: scegli l'istanza database di Amazon RDS che hai registrato in precedenza.
- Database name (Nome database) – Specificare il nome del database creato in precedenza, `nodeexampledb` per questo esempio.
- Repository type (Tipo di repository) - `Http Archive`.

  È necessario utilizzare questo tipo di repository per i file pubblici di Amazon S3. Il tipo `S3 Archive` viene utilizzato solo per gli archivi privati.
- URL del repository: l'URL Amazon S3 del file di archivio.

Utilizzare i valori predefiniti per le impostazioni restanti, quindi fare clic su Add App (Aggiungi app) per creare l'app.

4. Distribuire l'app.

   Andare alla pagina Apps (App) e nella colonna Actions (Operazioni) dell'app NodeDB scegliere deploy (distribuzione). Quindi scegli Deploy per distribuire l'app sulle istanze del server. AWS OpsWorks Stacks esegue le ricette Deploy su ogni istanza, che scarica l'applicazione dal repository e riavvia il server. Quando ogni istanza è contrassegnata da un segno di spunta verde e nel campo Status (Stato) è visualizzato successful (operazione riuscita), la distribuzione è completata e l'applicazione è pronta per inizializzare la gestione delle richieste.

   > **Note**
   >
   > Se la distribuzione ha esito negativo, scegliere show (mostra) nella colonna Log per visualizzare il log di Chef relativo alla distribuzione. Le informazioni relative agli errori si trovano verso il fondo.

5. Aprire l'applicazione .

   Per aprire l'applicazione, scegliere Layers (Livelli), scegliere il sistema di bilanciamento del carico, quindi scegliere il nome DNS di tale sistema. Questa operazione invia una richiesta HTTP al sistema di bilanciamento del carico. L'output visualizzato sarà simile a quanto segue.

AWS OpsWorks Node.js Example

Amazon RDS Endpoint: *nodeexample.cdlqlk5uwd0k.us-west-2.rds.amazonaws.com*

User: *opsworksuser*

Password: *Your-Pwd*

Port: *3306*

Database: *nodeexampledb*

Connection: successful

Databases: information_schema, innodb, mysql, nodeexampledb, performance_schema

Note

AWS OpsWorks Stacks distribuisce automaticamente le app su nuove istanze durante la configurazione. La distribuzione manuale è richiesta solo per le istanze online. Per ulteriori informazioni, consulta Distribuzione di app. Per una discussione generale sulla distribuzione, tra cui le strategie di distribuzione più avanzate, consulta Gestione e distribuzione di app e libri di ricette.

Come procedere

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

In questa procedura guidata sono state descritte le nozioni di base relative alla configurazione di un semplice stack del server di applicazioni Node.js. Di seguito sono elencati alcuni suggerimenti su cosa fare in seguito.

Esaminare i libri di ricette predefiniti di Node.js

Se vuoi sapere in dettaglio come sono configurate le istanze, consulta il ricettario integrato nel layer, opsworks_nodejs, che contiene le ricette e i file correlati utilizzati da AWS OpsWorks Stacks per installare e configurare il software, e il cookbook integrato di distribuzione, che contiene le ricette utilizzate da Stacks per distribuire le app. AWS OpsWorks

Personalizzare la configurazione del server

Lo stack di esempio è uno stack di base. Per l'uso in ambienti di produzione, potrebbe essere necessario personalizzare lo stack. Per ulteriori informazioni, consulta Personalizzazione degli stack AWS OpsWorks.

Aggiungere il supporto SSL

Puoi abilitare il supporto SSL per la tua app e fornire a Stacks i certificati appropriati quando crei l'app. AWS OpsWorks AWS OpsWorks Stacks installa quindi i certificati nella directory appropriata. Per ulteriori informazioni, consulta Uso di SSL.

Aggiungere il caching dei dati in memoria

I siti a livello di produzione spesso ottimizzano le prestazioni mediante il caching dei dati in uno store di chiave-valore in memoria, ad esempio Redis o Memcache. Puoi usarli entrambi con uno AWS OpsWorks stack Stacks. Per ulteriori informazioni, consultare ElastiCache Redis e Memcached.

Usare una strategia di distribuzione più complessa

Nell'esempio è stata utilizzata una strategia di distribuzione alquanto semplice, che prevede la distribuzione dell'aggiornamento in ogni istanza simultaneamente. Questo approccio è semplice e veloce, ma non vi è alcun margine di errore. Se la distribuzione ha esito negativo o l'aggiornamento genera problemi, ogni istanza nello stack di produzione potrebbe venire interessata e pertanto il sito potrebbe potenzialmente non funzionare o venire disabilitato finché non risolvi il problema. Per ulteriori informazioni sulle strategie di distribuzione, consulta Gestione e distribuzione di app e libri di ricette.

Estendi il livello Node.js App Server

Puoi estendere il livello in diversi modi. Ad esempio, puoi implementare ricette per eseguire script sulle istanze o implementare gli hook di distribuzione di Chef per personalizzare la distribuzione delle app. Per ulteriori informazioni, consulta Estensione di un livello.

Definire le variabili di ambiente

Puoi trasferire dati all'applicazione definendo variabili di ambiente per l'app associata. Quando distribuisci l'app, AWS OpsWorks Stacks esporta tali variabili in modo che tu possa accedervi dall'app. Per ulteriori informazioni, consulta Utilizzo delle variabili di ambiente .

Personalizzazione degli stack AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks I livelli integrati di Stack forniscono funzionalità standard sufficienti per molti scopi. Tuttavia, puoi trovarti di fronte a uno o più dei seguenti problemi:

- La configurazione standard di un livello integrato è adeguata ma non ideale e desideri ottimizzarla per i tuoi requisiti specifici.

  Ad esempio, potreste voler ottimizzare la configurazione del server Nginx del livello Static Web Server specificando i vostri valori per impostazioni come il numero massimo di processi di lavoro o il valore. `keepalivetimeout`
- Una funzionalità del livello integrato è corretta, ma desideri estenderla installando pacchetti aggiuntivi o eseguendo alcuni script d installazione personalizzati.

  Ad esempio, potresti voler estendere un livello PHP App Server installando anche un server Redis.
- I tuoi requisiti non sono gestiti da nessuno dei livelli integrati.

  Ad esempio, AWS OpsWorks Stacks non include livelli integrati per alcuni server di database più diffusi. Puoi creare un livello personalizzato che installa questi server nelle istanze del livello.
- Stai eseguendo uno stack Windows che supporta solo i livelli personalizzati.

AWS OpsWorks Stacks offre diversi modi per personalizzare i layer in base a requisiti specifici. I seguenti esempi sono elencati in ordine crescente relativamente alla complessità e alla potenza:

> **Note**
>
> Alcuni di questi approcci possono essere utilizzati solo per gli stack Linux. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Usa JSON personalizzato per sovrascrivere le impostazioni predefinite AWS OpsWorks degli Stacks.
- Implementa un ricettario Chef personalizzato con un file di attributi che sostituisce le impostazioni predefinite di Stacks. AWS OpsWorks
- Implementa un ricettario Chef personalizzato con un modello che sostituisce o estende un modello Stacks predefinito. AWS OpsWorks
- Implementa un libro di ricette Chef personalizzato con una ricetta semplice che esegue uno script della shell.
- Implementa un libro di ricette Chef personalizzato che esegue attività quali la creazione e la configurazione di directory, l'installazione di pacchetti, la creazione di file di configurazione, la distribuzione di app e così via.

Puoi inoltre sostituire ricette, a seconda del sistema operativo e della versione di Chef dello stack.

- Con gli stack Chef 0.9 e 11.4, non puoi sostituire una ricetta integrata implementando una ricetta personalizzata con lo stesso nome di ricetta e di libro di ricette.

  Per ogni evento del ciclo di vita, AWS OpsWorks Stacks esegue sempre per prime le ricette integrate, seguite da eventuali ricette personalizzate. Poiché queste versioni di Chef non eseguono due volte una ricetta con lo stesso nome di libro di ricette e di ricetta, la ricetta integrata avrà priorità sulla ricetta personalizzata, che non verrà di fatto eseguita.
- Puoi sostituire le ricette integrate negli stack Chef 11.10.

  Per ulteriori informazioni, consulta Installazione e precedenza dei libri di ricette.
- Le ricette integrate sugli stack Windows non possono essere sostituite.

  Il modo in cui AWS OpsWorks Stacks gestisce Chef run for Windows Stacks non consente di sovrascrivere le ricette integrate.

> **Note**
>
> Poiché molte delle tecniche utilizzano libri di cucina personalizzati, dovresti prima leggerli Libri di ricette e ricette se non hai già familiarità con l'implementazione dei libri di cucina. Nozioni di base sui libri di ricettefornisce un tutorial dettagliato sull'implementazione di libri di cucina personalizzati e Implementazione di Cookbooks for Stacks AWS OpsWorks illustra alcuni dettagli su come implementare i libri di cucina per le istanze di Stacks. AWS OpsWorks

Argomenti

- Personalizzazione della configurazione degli stack sovrascrivendo gli attributi AWS OpsWorks
- Estensione dei file di configurazione degli AWS OpsWorks stack utilizzando modelli personalizzati
- Estensione di un livello
- Creazione di un livello server Tomcat personalizzato
- Attributi di configurazione e distribuzione dello stack

Personalizzazione della configurazione degli stack sovrascrivendo gli attributi AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Per gli stack Windows e gli stack Chef 12 Linux, AWS OpsWorks Stacks utilizza esecuzioni Chef separate per ricette integrate e ricette personalizzate. Ciò significa che non puoi usare le tecniche descritte in questa sezione per sostituire gli attributi predefiniti degli stack di Windows e Chef 12 Linux.

Le ricette e i modelli dipendono da una serie di attributi di Chef relativi alle informazioni specifiche per le istanze o gli stack, ad esempio le configurazioni dei livelli o le impostazioni del server di applicazioni. Questi attributi hanno diverse origini:

- JSON personalizzato: puoi facoltativamente specificare attributi JSON personalizzati quando crei, aggiorni o cloni uno stack o quando distribuisci un'app.
- Attributi di configurazione dello stack:AWS OpsWorks Stacks definisce questi attributi per contenere le informazioni di configurazione dello stack, incluse le informazioni specificate tramite le impostazioni della console.
- Attributi di distribuzione: AWS OpsWorks definisce gli attributi relativi alla distribuzione per gli eventi Deploy.
- Attributi del libro di cucina: i Cookbook integrati e personalizzati di solito includono uno o più file di attributi, che contengono attributi che rappresentano valori specifici del libro di cucina, come le impostazioni di configurazione del server delle applicazioni.
- Lo strumento Ohai di Chef —Chef definisce attributi che rappresentano un'ampia varietà di impostazioni di configurazione del sistema, come il tipo di CPU e la memoria installata.

Per un elenco completo degli attributi di configurazione e distribuzione degli stack e degli attributi predefiniti dei libri di ricette, consulta Attributi di configurazione e distribuzione dello stack: Linux e Attributi dei libri di ricette predefiniti. Per ulteriori informazioni sugli attributi Ohai, consulta Ohai.

Quando si verifica un evento del ciclo di vita, ad esempio la distribuzione o la configurazione, oppure quando si esegue un comando dello stack, ad esempio `execute_recipes` o `update_packages`, AWS OpsWorks Stacks esegue le seguenti operazioni:

- Invia all'agente un comando corrispondente in ciascuna istanza interessata.

  L'agente esegue le ricette appropriate. Ad esempio, per un evento di distribuzione, l'agente esegue la ricetta di distribuzione predefinita, seguita da qualsiasi altra ricetta di distribuzione personalizzata.
- Unisce qualsiasi attributo JSON e di distribuzione personalizzato agli attributi di configurazione dello stack e li installa sulle istanze.

Gli attributi degli attributi personalizzati JSON e di configurazione e distribuzione dello stack, gli attributi del libro di ricette e gli attributi Ohai vengono uniti in un oggetto nodo, che fornisce i valori degli attributi alle ricette. Un'istanza è sostanzialmente stateless per quanto riguarda gli attributi

di configurazione dello stack, compreso qualsiasi attributo JSON personalizzato. Quando esegui un comando relativo alla distribuzione o allo stack, le ricette associate utilizzano gli attributi di configurazione dello stack scaricati con il comando.

Argomenti

- Precedenza degli attributi
- Sostituzione di attributi in formato JSON personalizzato
- Sovrascrivere gli attributi Stacks utilizzando gli attributi personalizzati AWS OpsWorks del cookbook

Precedenza degli attributi

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se un attributo è definito in modo univoco, Chef lo incorpora nell'oggetto nodo. Tuttavia, qualsiasi origine di attributo può definire qualsiasi attributo. Pertanto lo stesso attributo può avere più definizioni con valori diversi. Ad esempio, il libro di ricette `apache2` predefinito definisce `node[:apache][:keepalive]`, ma puoi anche definire tale attributo in formato JSON personalizzato o in un libro di ricette personalizzato. Se un attributo ha più definizioni, queste vengono valutate nell'ordine descritto più avanti; l'oggetto nodo riceve la definizione caratterizzata dalla precedenza massima.

Un attributo viene definita nel seguente modo:

```
node.type[:attribute][:sub_attribute][:...]=value
```

Se un attributo ha più definizioni, il tipo determina quale definizione ha la precedenza e tale definizione viene incorporata nell'oggetto nodo. AWS OpsWorks Stacks utilizza i seguenti tipi di attributi:

- default: questo è il tipo più comune e significa essenzialmente «usa questo valore se l'attributo non è già stato definito». Se tutte le definizioni di un attributo sono di tipo `default`, la prima definizione

nell'ordine di valutazione avrà la priorità, mentre i valori successivi verranno ignorati. Nota che AWS OpsWorks Stacks imposta tutte le definizioni degli attributi di configurazione e distribuzione dello stack su type. `default`

- normal: gli attributi di questo tipo sostituiscono tutti `default` `normal` gli attributi definiti in precedenza nell'ordine di valutazione. Ad esempio, se il primo attributo proviene da un libro di ricette predefinito ed è associato al tipo `default`, mentre il secondo attributo è un attributo definito dall'utente ed è associato al tipo `normal`, la seconda definizione avrà la precedenza.
- set —Questo è un tipo obsoleto che potresti vedere nei vecchi libri di cucina. È stato sostituito da `normal`, che è caratterizzato dalla stessa priorità.

Chef supporta vari tipi di attributo aggiuntivi, tra cui il tipo `automatic`, che ha la precedenza su tutte le altre definizioni di attributo. Le definizioni di attributo generate dallo strumento Ohai di Chef sono tutte di tipo `automatic`. Sono pertanto di sola lettura. Di solito questo non è un problema, perché non c'è motivo di sostituirli e sono distinti dagli attributi di Stacks. AWS OpsWorks Tuttavia, è consigliabile prestare particolare attenzione quando gli attributi personalizzati del libro di ricette vengono rinominati affinché siano effettivamente diversi dagli attributi Ohai. Per ulteriori informazioni, consulta la sezione relativa alle informazioni sugli attributi.

> **Note**
>
> Lo strumento Ohai è un eseguibile che puoi eseguire dalla riga di comando. Per elencare gli attributi Ohai di un'istanza, accedi all'istanza ed esegui `ohai` in una finestra del terminale. Devi considerare il fatto che questo comando genera un output di grandi dimensioni.

Di seguito sono descritti i passaggi che consentono di incorporare le diverse definizioni di attributo nell'oggetto nodo:

1. Unire gli attributi di configurazione personalizzati dello stack negli attributi di configurazione e distribuzione dello stack.

   Gli attributi JSON personalizzati possono essere impostati per lo stack o per una distribuzione specifica. Occupano sempre la prima posizione nell'ordine di valutazione in quanto sono di tipo `normal`. Se anche uno o più attributi di configurazione dello stack vengono definiti in formato JSON personalizzato, i valori JSON personalizzati hanno la precedenza. In caso contrario, AWS OpsWorks Stacks si limita a incorporare gli attributi JSON personalizzati nella configurazione dello stack.

2. Unire gli attributi JSON di distribuzione personalizzati negli attributi di configurazione e distribuzione dello stack.

   Gli attributi JSON di distribuzione personalizzati sono di tipo `normal` e pertanto hanno la precedenza rispetto agli attributi JSON predefiniti e personalizzati di configurazione dello stack e rispetto agli attributi JSON di distribuzione predefiniti.
3. Unire gli attributi di configurazione e implementazione dello stack nell'oggetto nodo dell'istanza.
4. Unire gli attributi predefiniti del libro di ricette dell'istanza nell'oggetto nodo.

   Gli attributi predefiniti del libro di ricette sono tutti di tipo `default`. Se uno o più attributi incorporati del libro di cucina sono definiti anche negli attributi di configurazione e distribuzione dello stack, in genere perché li hai definiti con un codice JSON personalizzato, le definizioni di configurazione dello stack hanno la precedenza sulle definizioni integrate del libro di cucina. Tutti gli altri attributi predefiniti dei libri di ricette vengono semplicemente incorporati nell'oggetto nodo.
5. Unire gli attributi personalizzati del libro di ricette dell'istanza nell'oggetto nodo.

   Gli attributi personalizzati dei libri di ricette sono in genere di tipo `normal` o `default`. Gli attributi univoci sono incorporati nell'oggetto nodo. Se nei passaggi 1—3 sono definiti anche degli attributi personalizzati del libro di cucina (in genere perché li hai definiti con JSON personalizzato), la precedenza dipende dal tipo di attributo del libro di cucina personalizzato:

   - Gli attributi definiti nei passaggi 1—3 hanno la precedenza sugli attributi personalizzati del libro di cucina. `default`
   - `normal`Gli attributi personalizzati del cookbook hanno la precedenza sulle definizioni dei passaggi 1—3.

> **Important**
>
> Non utilizzare attributi personalizzati di `default` del libro di ricette per sostituire gli attributi di configurazione dello stack o gli attributi predefiniti del libro di ricette. Poiché gli attributi personalizzati del libro di ricette vengono valutati per ultimi, gli attributi di tipo `default` hanno la priorità più bassa e non possono sostituire nulla.

Sostituzione di attributi in formato JSON personalizzato

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Poiché AWS OpsWorks Stacks gestisce Chef funziona in modo diverso per gli stack Windows rispetto agli stack Linux, non è possibile utilizzare le tecniche illustrate in questa sezione per gli stack di Windows.

Il modo più semplice per sovrascrivere un attributo AWS OpsWorks Stacks è definirlo in JSON personalizzato, che ha la precedenza sugli attributi di configurazione e distribuzione dello stack, nonché sugli attributi di cookbook integrati e personalizzati. `default` Per ulteriori informazioni, consulta Precedenza degli attributi.

> **Important**
>
> È consigliabile sostituire gli attributi di configurazione e distribuzione dello stack prestando particolare attenzione. Ad esempio, la sostituzione degli attributi nello spazio dei nomi `opsworks` potrebbe interferire con le ricette predefinite. Per ulteriori informazioni, consulta Attributi di configurazione e distribuzione dello stack.

Puoi anche utilizzare il formato JSON personalizzato per definire gli attributi univoci, in genere per passare i dati alle ricette personalizzate. Gli attributi vengono incorporato nell'oggetto nodo e le ricette possono fare riferimento a essi mediante l'utilizzo della sintassi standard del nodo Chef.

Come specificare un formato JSON personalizzato

Per usare il formato JSON personalizzato per sostituire un valore di attributo, devi prima determinare il nome completo dell'attributo. Procederai quindi a creare un oggetto JSON che contiene gli

attributi che desideri sostituire, impostati sui valori preferiti. Per comodità, i documenti Attributi di configurazione e distribuzione dello stack: Linux e Attributi dei libri di ricette predefiniti usano gli attributi di configurazione, distribuzione e predefiniti dello stack più comunemente utilizzati, compresi i relativi nomi completi.

Le relazioni padre-figlio dell'oggetto devono corrispondere ai nodi Chef completi appropriati. Supponiamo, ad esempio, che tu voglia modificare i seguenti attributi Apache:

- Attributo `keepalivetimeout`, il cui nodo è `node[:apache][:keepalivetimeout]` e ha il valore predefinito di `3`.
- Attributo `logrotate` `schedule`, il cui nodo è `node[:apache][:logrotate][:schedule]` e ha il valore predefinito di `"daily"`.

Per sostituire gli attributi e impostare i valori rispettivamente su `5` e `"weekly"`, utilizza il seguente formato JSON personalizzato:

```
{
  "apache" : {
    "keepalivetimeout" : 5,
    "logrotate" : {
       "schedule" : "weekly"
    }
  }
}
```

Quando specificare un formato JSON personalizzato

Puoi specificare una struttura JSON personalizzata per le attività seguenti:

- Creare un nuovo stack
- Aggiornare uno stack
- Eseguire un comando dello stack
- Clonare uno stack
- Distribuire un'app

Per ogni attività, AWS OpsWorks Stacks unisce gli attributi JSON personalizzati con gli attributi di configurazione e distribuzione dello stack e li invia alle istanze, per essere uniti nell'oggetto nodo. Si noti, tuttavia, quanto segue:

- Se specifichi un formato JSON personalizzato quando crei, cloni o aggiorni uno stack, gli attributi vengono uniti agli attributi di configurazione e distribuzione dello stack per tutti gli eventi del ciclo di vita e per tutti i comandi dello stack successivi.
- Se specifichi un formato JSON personalizzato per una distribuzione, gli attributi vengono uniti agli attributi di configurazione e distribuzione dello stack solo per l'evento corrispondente.

  Se vuoi utilizzare questi attributi personalizzati per distribuzioni successive, devi specificare di nuovo il formato JSON personalizzato in modo esplicito.

È importante ricordare che gli attributi interessano l'istanza solo quando vengono usati dalle ricette. Se sostituisci un valore di attributo ma nessuna ricetta successiva fa riferimento all'attributo, la modifica non ha alcun effetto. Devi assicurarti che il formato JSON personalizzato venga inviato prima dell'esecuzione delle ricette associate oppure che le ricette appropriate vengano eseguite nuovamente.

Best practice relative al formato JSON personalizzato

Puoi utilizzare JSON personalizzato per sovrascrivere qualsiasi attributo AWS OpsWorks Stacks, ma l'immissione manuale delle informazioni è piuttosto complicata e non è soggetta a alcun tipo di controllo del codice sorgente. Il formato JSON personalizzato è particolarmente indicato nei seguenti scenari:

- Quando vuoi sostituire solo un numero limitato di attributi e non devi necessariamente utilizzare i libri di ricette personalizzati.

  Grazie al formato JSON personalizzato puoi evitare di configurare e gestire un archivio di libri di ricette per sostituire solo un paio di attributi.
- In caso di utilizzo di valori sensibili, ad esempio password o chiavi di autenticazione.

  Gli attributi dei libri di ricette vengono archiviati in un archivio e pertanto le informazioni sensibili potrebbero venire compromesse. È pertanto consigliabile definire gli attributi con valori fittizi e utilizzare il formato JSON personalizzato per impostare i valori reali.
- In caso di utilizzo di valori che possono variare.

Ad esempio, è consigliabile disporre di uno stack di produzione supportato da stack di sviluppo e gestione temporanea distinti. Supponiamo che questi stack supportino un'applicazione che accetta pagamenti. Se utilizzi il formato JSON personalizzato per specificare l'endpoint di pagamento, puoi specificare un URL di test per lo stack di gestione temporanea. Quando sei pronto a eseguire la migrazione di uno stack aggiornato nello stack di produzione, puoi utilizzare gli stessi libri di ricette e utilizzare il formato JSON personalizzato per impostare l'endpoint di pagamento per l'URL di produzione.

- In caso di utilizzo di valori specifici per un determinato comando di implementazione o stack.

Sovrascrivere gli attributi Stacks utilizzando gli attributi personalizzati AWS OpsWorks del cookbook

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Per gli stack Windows, AWS OpsWorks Stacks utilizza esecuzioni Chef separate per ricette integrate e ricette personalizzate. Ciò significa che non puoi usare le tecniche descritte in questa sezione per sostituire gli attributi predefiniti degli stack di Windows.

Il codice JSON personalizzato è un modo conveniente per sovrascrivere la configurazione dello stack di AWS OpsWorks Stacks e gli attributi incorporati del libro di cucina, ma presenta alcune limitazioni. In particolare, devi immettere manualmente il formato JSON personalizzato a ogni utilizzo. Per tale motivo, non disponi di un metodo affidabile per gestire le definizioni. Un approccio migliore è spesso rappresentato dai file di attributi personalizzati del libro di ricette per sostituire gli attributi predefiniti. In questo modo le definizioni saranno gestite a livello di codice sorgente.

La procedura per utilizzare i file di attributi personalizzati per AWS OpsWorks sovrascrivere le definizioni di Stacks è semplice.

Per sovrascrivere le definizioni degli attributi Stacks AWS OpsWorks

1. Configurare un archivio di libri di ricette, come descritto nell'argomento Libri di ricette e ricette.
2. Creare un libro di ricette con lo stesso nome del libro di ricette predefinito, contenente gli attributi che si desidera sostituire. Ad esempio, per sostituire gli attributi di Apache, il libro di ricette deve essere denominato apache2.
3. Aggiungere una cartella `attributes` al libro di ricette e aggiungere un file denominato `customize.rb` a tale cartella.
4. Aggiungere una definizione di attributo al file per ciascun attributo predefinito del libro di ricette che si desidera sostituire e impostarlo sul valore desiderato. L'attributo deve essere di `normal` tipo o superiore e avere esattamente lo stesso nome di nodo dell'attributo AWS OpsWorks Stacks corrispondente. Per un elenco dettagliato degli attributi AWS OpsWorks Stacks, inclusi i nomi dei nodi, consulta Attributi di configurazione e distribuzione dello stack: Linux e. Attributi dei libri di ricette predefiniti Per ulteriori informazioni su attributi e file di attributi, consulta l'argomento relativo alle informazioni sui file di attributi.

   > **Important**
   >
   > Gli attributi devono essere `normal` di tipo per sostituire gli attributi AWS OpsWorks Stacks; i `default` tipi non hanno la precedenza. Ad esempio, se il file `customize.rb` contiene una definizione di attributo `default[:apache][:keepalivetimeout] = 5`, l'attributo corrispondente nel file di attributi predefiniti `apache.rb` viene valutato per primo in quanto ha la precedenza. Per ulteriori informazioni, consulta Sostituzione degli attributi.

5. Ripeti i passaggi da 2 a 4 per ogni ricettario integrato con gli attributi che desideri sovrascrivere.
6. Abilita i libri di cucina personalizzati per il tuo stack e fornisci le informazioni necessarie a AWS OpsWorks Stacks per scaricare i tuoi libri di cucina nelle istanze dello stack. Per ulteriori informazioni, consulta Installazione di libri di ricette personalizzati.

> **Note**
>
> Per una procedura guidata completa relativa a questa procedura, consulta Sostituzione degli attributi integrati.

L'oggetto node utilizzato dai successivi eventi del ciclo di vita, dai comandi deploy e dai comandi stack ora conterrà le definizioni degli attributi anziché i valori Stacks. AWS OpsWorks

Ad esempio, per sostituire le impostazioni Apache `keepalivetimeout` e `logrotate schedule` predefinite discusse in Come specificare un formato JSON personalizzato, aggiungere un libro di ricette `apache2` all'archivio e quindi aggiungere un file `customize.rb` alla cartella `attributes` del libro di ricette contenente il seguente contenuto.

```
normal[:apache][:keepalivetimeout] = 5
normal[:apache][:logrotate][:schedule] = 'weekly'
```

Important

Non è necessario sovrascrivere gli attributi AWS OpsWorks Stacks modificando una copia del file degli attributi incorporati associato. Se, ad esempio, copi `apache.rb` nella cartella `apache2/attributes` e modifichi alcune impostazioni, non fai altro che sostituire ciascun attributo nel file predefinito. Le ricette utilizzeranno le definizioni di attributo incluse nella copia e ignoreranno il file predefinito. Se in seguito AWS OpsWorks Stacks modifica il file di attributi predefiniti, le ricette non avranno accesso alle modifiche a meno che tu non aggiorni manualmente la copia.
Per evitare questa situazione, tutti i libri di ricette predefiniti contengono un file di attributi `customize.rb` vuoto, che è richiesto in tutti i moduli tramite una direttiva `include_attribute`. La sostituzione degli attributi nella copia di `customize.rb` interessa solo tali attributi specifici. Le ricette recupereranno gli altri valori di attributo dai file di attributi predefiniti e otterranno automaticamente i valori correnti di qualsiasi attributo non sostituito. Grazie a questo approccio puoi ridurre il numero di attributi presenti nell'archivio di libri di ricette e pertanto ridurre l'impegno necessario per la manutenzione, nonché semplificare la gestione degli aggiornamenti futuri.

Estensione dei file di configurazione degli AWS OpsWorks stack utilizzando modelli personalizzati

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Poiché AWS OpsWorks Stacks gestisce Chef funziona in modo diverso per gli stack Windows rispetto agli stack Linux, non è possibile utilizzare le tecniche illustrate in questa sezione per gli stack di Windows.

AWS OpsWorks Stacks utilizza modelli per creare file come i file di configurazione, che in genere dipendono dagli attributi di molte impostazioni. Se utilizzi attributi JSON personalizzati o personalizzati del libro di cucina per sovrascrivere le definizioni di AWS OpsWorks Stacks, le tue impostazioni preferite vengono incorporate nei file di configurazione al posto delle impostazioni Stacks. AWS OpsWorks Tuttavia, AWS OpsWorks Stacks non specifica necessariamente un attributo per ogni possibile impostazione di configurazione; accetta i valori predefiniti per alcune impostazioni e ne codifica altre direttamente nel modello. Non puoi utilizzare attributi JSON personalizzati o attributi personalizzati del libro di cucina per specificare le impostazioni preferite se non esiste un attributo Stacks corrispondente. AWS OpsWorks

È possibile estendere il file di configurazione per includere impostazioni di configurazione aggiuntive creando un modello personalizzato. È quindi possibile aggiungere al file qualsiasi impostazione di configurazione o qualsiasi altro contenuto di cui si ha bisogno e sostituire le impostazioni hardcoded. Per ulteriori informazioni sui modelli, consulta Modelli.

> **Note**
>
> È possibile sostituire qualsiasi modello incorporato eccetto opsworks-agent.monitrc.erb.

Per creare un modello personalizzato

1. Creare un libro di ricette con la stessa struttura e gli stessi nomi di directory del libro di ricette incorporato. Quindi, creare un file di modello nella directory appropriata con lo stesso nome del modello incorporato che si desidera personalizzare. Per esempio, per utilizzare un modello personalizzato per estendere il file di configurazione Apache `httpd.conf`, è necessario implementare un libro di ricette `apache2` nel repository e i file di modello devono essere

`apache2/templates/default/apache.conf.erb`. L'uso degli stessi nomi consente a AWS OpsWorks Stacks di riconoscere il modello personalizzato e di utilizzarlo al posto del modello integrato.

L'approccio più semplice consiste nel copiare semplicemente il file modello integrato dall' GitHubarchivio del ricettario integrato nel ricettario e modificarlo secondo necessità.

> **Important**
>
> Non copiare i file dal libro di ricette incorporato ad eccezione dei file di modello che si desidera personalizzare. Le copie di altre tipologie di file del libro di ricette, come per esempio le ricette, creano risorse Chef duplicate e possono causare errori.

Il libro di ricette può anche includere attributi personalizzati, ricette e file correlati, ma i nomi dei loro file non devono duplicare i nomi del file incorporato.

2. Personalizza il file di modello per ottenere un file di configurazione che soddisfa i tuoi requisiti. È possibile aggiungere ulteriori impostazioni, eliminare le impostazioni esistenti, sostituire gli attributi hardcoded e così via.
3. Se ancora non è stato fatto, modificare le impostazioni dello stack per abilitare i libri di ricette personalizzati e specificare la repository del libro di ricette. Per ulteriori informazioni, consulta Installazione di libri di ricette personalizzati.

> **Note**
>
> Per una procedura guidata completa relativa a questa procedura, consulta Sostituzione dei modelli integrati.

Non è necessario implementare alcuna ricetta o aggiungere ricette alla configurazione dei livelli per sovrascrivere un modello. AWS OpsWorks Stacks esegue sempre le ricette integrate. Quando esegue la ricetta che crea il file di configurazione, utilizzerà automaticamente il modello personalizzato anziché il modello incorporato.

> **Note**
>
> Se AWS OpsWorks Stacks apporta modifiche al modello integrato, il modello personalizzato potrebbe non essere sincronizzato e non funzionare più correttamente. Ad esempio, supponiamo che il modello faccia riferimento a un file dipendente e che il nome del file cambi. AWS OpsWorks Stacks non apporta tali modifiche spesso e, quando un modello cambia, elenca le modifiche e ti offre la possibilità di eseguire l'aggiornamento a una nuova versione. È necessario monitorare l'archivio AWS OpsWorks Stacks per rilevare eventuali modifiche e aggiornare manualmente il modello in base alle esigenze.

Estensione di un livello

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

A volte, è necessario personalizzare un livello incorporato al di là di quanto può essere gestito modificando gli attributi AWS OpsWorks Stacks o personalizzando i modelli. Per esempio, supponiamo la necessità di creare dei symlink, impostare il file o le modalità della cartella, installare dei pacchetti aggiuntivi e così via. È necessario estendere i livelli personalizzati per fornire funzionalità più che minime. In questo caso, è necessario implementare uno o più libri di ricette personalizzati con ricette per gestire le attività di personalizzazione. Questo argomento fornisce alcuni esempi di come utilizzare le ricette per estendere un livello.

Se utilizzi Chef per la prima volta, leggi Introduzione ai libri di ricette, un tutorial che introduce i concetti di base sull'implementazione di libri di ricette e l'esecuzione di varie attività comuni. Per un esempio dettagliato di come implementare un livello personalizzato, consulta Creazione di un livello server Tomcat personalizzato.

Argomenti

- Utilizzo delle ricette per eseguire gli script
- Utilizzo degli hook di distribuzione di Chef

- Esecuzione di processi Cron sulle istanze Linux
- Installazione e configurazione di pacchetti sulle istanze Linux

Utilizzo delle ricette per eseguire gli script

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se si dispone già di uno script che esegue le attività di personalizzazione necessarie, l'approccio più semplice per estendere un livello è spesso quello di implementare una semplice ricetta per eseguire lo script. È quindi possibile assegnare la ricetta agli eventi del ciclo di vita appropriati, in genere Configurazione o Distribuzione, oppure utilizzare il comando stack `execute_recipes` per eseguire la ricetta manualmente.

L'esempio seguente esegue uno script di shell su istanze Linux, ma puoi usare lo stesso approccio per altri tipi di script, inclusi gli script di Windows. PowerShell

```
cookbook_file "/tmp/lib-installer.sh" do
  source "lib-installer.sh"
  mode 0755
end

execute "install my lib" do
  command "sh /tmp/lib-installer.sh"
end
```

La risorsa `cookbook_file` rappresenta un file archiviato in una sottodirectory della directory `files` di un libro di ricette e trasferisce il file in un percorso specificato sull'istanza. Questo esempio trasferisce uno script della shell, `lib-installer.sh`, nella directory `/tmp` dell'istanza e imposta la modalità del file in `0755`. Per ulteriori informazioni, consulta cookbook_file.

La risorsa `execute` rappresenta un comando, come per esempio un comando della shell. Questo esempio esegue `lib-installer.sh`. Per ulteriori informazioni, consulta execute.

È anche possibile eseguire uno script incorporandolo in una ricetta. L'esempio seguente esegue uno script bash, ma Chef supporta anche Csh, Perl, Python e Ruby.

```
script "install_something" do
  interpreter "bash"
  user "root"
  cwd "/tmp"
  code <<-EOH
    #insert bash script
  EOH
end
```

La risorsa `script` rappresenta uno script. L'esempio specifica un interprete bash, imposta l'utente su `"root"`e imposta la directory di lavoro su `/tmp`. Quindi esegue lo script bash nel blocco `code`, che può includere il numero di righe necessario. Per ulteriori informazioni, consulta script.

Per ulteriori informazioni su come utilizzare le ricette per eseguire gli script, consulta Esempio 7: esecuzione di comandi e script. Per un esempio di come eseguire PowerShell uno script su un'istanza di Windows, vedi. Esecuzione di uno PowerShell script di Windows

Utilizzo degli hook di distribuzione di Chef

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

È possibile personalizzare la distribuzione implementando una ricetta personalizzata per eseguire le attività necessarie e assegnandola all'evento di distribuzione appropriato del livello. Un approccio alternativo e talvolta più semplice, specialmente se non è necessario implementare un libro di cucina per altri scopi, consiste nell'utilizzare gli hook di distribuzione di Chef per eseguire il codice

di personalizzazione. Inoltre, le ricette di distribuzione personalizzate vengono eseguite dopo che la distribuzione è già stata eseguita dalle ricette incorporate. Gli hook di distribuzione consentono di interagire durante una distribuzione, per esempio, dopo che il codice dell'app viene estratto dal repository, ma prima che Apache venga riavviato.

Chef distribuisce le app in quattro fasi:

- Checkout: scarica i file dal repository
- Migra: esegue una migrazione, come richiesto
- Symlink: crea collegamenti simbolici
- Riavvia: riavvia l'applicazione

Gli hook di distribuzione di Chef forniscono un modo semplice per personalizzare una distribuzione tramite l'esecuzione facoltativa di un'applicazione Ruby fornita dall'utente dopo che ciascuna fase viene completata. Per utilizzare gli hook di distribuzione, implementare una o più applicazioni Ruby e posizionarle nella directory `/deploy` dell'app. Se l'applicazione non ha una directory `/deploy`, crearne una nel livello `APP_ROOT`. L'applicazione deve disporre di uno dei seguenti nomi, che determina quando è in esecuzione.

- `before_migrate.rb` viene eseguito dopo il completamento della fase Checkout, ma prima di Migrate.
- `before_symlink.rb` viene eseguito dopo il completamento della fase Migrate, ma prima di Symlink.
- `before_restart.rb` viene eseguito dopo il completamento della fase Symlink, ma prima di Restart.
- `after_restart.rb` viene eseguito dopo il completamento della fase Restart.

Gli hook di distribuzione di Chef possono accedere all'oggetto nodo utilizzando una sintassi di nodo standard, come avviene per le ricette. Gli hook di distribuzione possono anche accedere ai valori di qualsiasi variabile di ambiente di app specificata. Tuttavia, è necessario utilizzare `new_resource.environment["VARIABLE_NAME"]` per accedere al valore della variabile anziché `ENV["VARIABLE_NAME"]`.

Esecuzione di processi Cron sulle istanze Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Un processo Cron di Linux indica al daemon Cron di eseguire uno o più comandi su una pianificazione specificata. Per esempio, supponiamo che lo stack supporti un'applicazione e-commerce PHP. È possibile impostare un processo Cron per fare in modo che il server invii un report delle vendite in un orario specificato ogni settimana. Per ulteriori informazioni su Cron, consulta cron su Wikipedia. Per ulteriori informazioni su come eseguire un processo Cron direttamente su un computer basato su Linux o su un'istanza, consulta l'argomento che spiega che cosa sono Cron e Crontab e come si utilizzano sul sito Web delle conoscenze di base dell'Università dell'Indiana.

Anche se è possibile impostare manualmente i processi `cron` sulle istanze singole basate su Linux collegandosi a esse con SSH e modificando le loro voci `crontab`, uno dei principali vantaggi di AWS OpsWorks Stacks è che è possibile indicargli di eseguire l'attività su un intero livello di istanze. La procedura seguente descrive come configurare un `cron` lavoro sulle istanze di un livello di PHP App Server, ma è possibile utilizzare lo stesso approccio con qualsiasi livello.

Per configurare un processo **`cron`** sulle istanze di un livello

1. Implementare un libro di ricette con una ricetta con una risorsa `cron` che imposta il processo. L'esempio presuppone che la ricetta sia denominata `cronjob.rb`; i dettagli di implementazione vengono descritti in seguito. Per ulteriori informazioni sui libri di ricette e sulle ricette, consulta Libri di ricette e ricette.
2. Installare il libro di ricette sullo stack. Per ulteriori informazioni, consulta Installazione di libri di ricette personalizzati.
3. Fai in modo che AWS OpsWorks Stacks esegua automaticamente la ricetta sulle istanze del layer assegnandola ai seguenti eventi del ciclo di vita. Per ulteriori informazioni, consulta Esecuzione automatica di ricette.

- Configurazione: l'assegnazione `cronjob.rb` a questo evento indica a AWS OpsWorks Stacks di eseguire la ricetta su tutte le nuove istanze.
- Distribuisci: l'assegnazione `cronjob.rb` a questo evento indica a AWS OpsWorks Stacks di eseguire la ricetta su tutte le istanze online quando distribuisci o ridistribuisci un'app sul layer.

È anche possibile eseguire manualmente la ricetta sulle istanze online utilizzando il comando stack `Execute Recipes`. Per ulteriori informazioni, consulta Esecuzione dei comandi dello stack.

Di seguito è riportato l'esempio `cronjob.rb`, che imposta un processo Cron per eseguire un'applicazione PHP implementata dall'utente una volta a settimana, la quale raccoglie i dati di vendita dal server e invia un report per email. Per ulteriori esempi di come utilizzare una risorsa Cron, consulta cron.

```
cron "job_name" do
  hour "1"
  minute "10"
  weekday "6"
  command "cd /srv/www/myapp/current && php .lib/mailing.php"
end
```

`cron` è una risorsa Chef che rappresenta un processo `cron`. Quando AWS OpsWorks Stacks esegue la ricetta su un'istanza, il provider associato gestisce i dettagli della configurazione del lavoro.

- *`job_name`* è un nome definito dall'utente per il processo `cron`, come per esempio `weekly report`.
- `hour`/`minute`/`weekday` specificano quando i comandi devono essere eseguiti. Questo esempio esegue i comandi ogni sabato alle 01:10.
- `command` specifica i comandi da eseguire.

  Questo esempio esegue due comandi. Il primo passa alla directory `/srv/www/myapp/current`. Il secondo esegue l'applicazione `mailing.php` implementata dall'utente, che raccoglie i dati di vendita e invia il report.

> **Note**
>
> Il comando `bundle` non funziona con i processi `cron` per impostazione predefinita. Il motivo è che AWS OpsWorks Stacks installa bundler nella directory. `/usr/local/bin` Per utilizzare `bundle` con un processo `cron`, bisogna aggiungere esplicitamente il percorso `/usr/local/bin` al processo Cron. Inoltre, poiché la variabile di ambiente $PATH potrebbe non espandersi nel processo `cron`, una best practice è quella di aggiungere esplicitamente qualsiasi informazione sul percorso necessaria al processo senza dover ricorrere all'espansione della variabile $PATH. Gli esempi seguenti mostrano due modi per utilizzare `bundle` in un processo `cron`.
>
> ```
> cron "my first task" do
>   path "/usr/local/bin"
>   minute "*/10"
>   command "cd /srv/www/myapp/current && bundle exec my_command"
> end
> ```
>
> ```
> cron_env = {"PATH" => "/usr/local/bin"}
> cron "my second task" do
>   environment cron_env
>   minute "*/10"
>   command "cd /srv/www/myapp/current && /usr/local/bin/bundle exec my_command"
> end
> ```

Se lo stack ha più server di applicazioni, l'assegnazione degli eventi del ciclo `cronjob.rb` di vita del livello PHP App Server potrebbe non essere l'approccio ideale. Per esempio, la ricetta viene eseguita su tutte le istanze del livello, quindi verranno ricevuti più report. Un approccio migliore è quello di utilizzare un livello personalizzato per assicurare che solo un server invii un report.

Per eseguire una ricetta solo su una delle istanze di un livello

1. Crea un layer personalizzato chiamato, ad esempio, PHPAdmin e assegnalo ai relativi eventi Setup and Deploy`cronjob.rb`. I livelli personalizzati non devono necessariamente fare molto. In questo caso, esegue PHPAdmin solo una ricetta personalizzata sulle relative istanze.

2. Assegna una delle istanze di PHP App Server a. AdminLayer Se un'istanza appartiene a più di un livello, AWS OpsWorks Stacks esegue le ricette integrate e personalizzate di ogni livello.

Poiché solo un'istanza appartiene al PHP App Server e ai PHPAdmin livelli, `cronjob.rb` viene eseguita solo su quell'istanza e si riceve un solo rapporto.

Installazione e configurazione di pacchetti sulle istanze Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

I livelli incorporati supportano solo determinati pacchetti. Per ulteriori informazioni, consulta Livelli. È possibile installare altri pacchetti, come un server Redis, implementando le ricette personalizzate affinché gestiscano l'installazione associata, la configurazione e le attività di distribuzione. In alcuni casi, l'approccio migliore è quello di estendere un livello incorporato affinché installi il pacchetto sulle sue istanze insieme ai pacchetti standard del livello. Ad esempio, se disponi di uno stack che supporta un'applicazione PHP e desideri includere un server Redis, puoi estendere il livello PHP App Server per installare e configurare un server Redis sulle istanze del livello oltre a un server di applicazioni PHP.

Una ricetta di installazione di un pacchetto in genere deve effettuare attività come queste:

- Creare una o più directory e impostare le loro modalità.
- Creare un file di configurazione da un modello.
- Eseguire il programma di installazione per installare il pacchetto nell'istanza.
- Avviare uno o più servizi.

Per un esempio di come installare un server Tomcat, consulta Creazione di un livello server Tomcat personalizzato. L'argomento descrive come impostare un livello Redis personalizzato, ma è possibile utilizzare lo stesso codice per installare e configurare Redis su un livello incorporato. Per esempi di

come installare altri pacchetti, consulta i libri di cucina integrati, all'indirizzo opsworks-cookbooks. https://github.com/aws/

Creazione di un livello server Tomcat personalizzato

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> In questo argomento viene descritto come implementare un livello personalizzato per uno stack Linux. Tuttavia, i principi di base e parte del codice possono anche essere adattati allo scopo di implementare livelli personalizzati per gli stack Windows, soprattutto quelli nella sezione relativa alla distribuzione di app.

Il modo più semplice per utilizzare pacchetti non standard sulle istanze AWS OpsWorks Stacks consiste nell'estendere un livello esistente. Questo approccio però comporta l'installazione e l'esecuzione sia di pacchetti standard sia di pacchetti non standard sulle istanze del livello, creando una situazione non sempre auspicabile. Un approccio più complesso, ma più efficace, consiste nell'implementazione di un livello personalizzato, in grado di offrirti un controllo quasi completo delle istanze del livello, incluse quelle indicate di seguito:

- Pacchetti installati
- Configurazione di ogni pacchetto
- Distribuzione di app da un repository sull'istanza

Utilizzando la console o l'API, puoi creare e gestire un livello personalizzato come qualsiasi altro livello, come descritto in Livelli personalizzati. Tuttavia, le ricette incorporate di un livello personalizzato eseguono solo alcune attività assolutamente di base, come l'installazione di un client Ganglia per la segnalazione dei parametri a un master Ganglia. Per rendere le istanze di un

livello personalizzato più che minimamente funzionali, devi implementare uno o più libri di ricette personalizzati con ricette Chef e file correlati per gestire le attività di installazione e configurazione di pacchetti, distribuzione di app e così via. Tuttavia, non devi necessariamente implementare tutto da zero. Ad esempio, se archivi applicazioni in uno dei repository standard, puoi utilizzare le ricette di distribuzione integrate per gestire la maggior parte del lavoro di installazione delle applicazioni sulle istanze del livello.

> **Note**
>
> Se utilizzi Chef per la prima volta, leggi Introduzione ai libri di ricette, un tutorial che introduce i concetti di base sull'implementazione di libri di ricette e l'esecuzione di varie attività comuni.

Le seguenti procedure guidate illustrano come implementare un livello personalizzato che supporta un server di applicazioni Tomcat. Il livello si basa su un libro di ricette personalizzato denominato Tomcat che include, tra l'altro, ricette per gestire l'installazione e la distribuzione di pacchetti. La procedura guidata include estratti del libro di ricette Tomcat. Puoi scaricare il ricettario completo dal suo repository. GitHub Se non conosci Opscode Chef, leggi Libri di ricette e ricette.

> **Note**
>
> AWS OpsWorks Stacks include un livello Java App Server completo per l'uso in produzione. Lo scopo del libro di ricette Tomcat è illustrare come implementare livelli personalizzati, pertanto supporta solo una versione limitata di Tomcat, che non include caratteristiche come il protocollo SSL. Per ottenere un esempio di implementazione completa, consulta il libro di ricette opsworks_java integrato.

Il libro di ricette Tomcat supporta un livello personalizzato le cui istanze presentano le seguenti caratteristiche:

- Supportano un server di applicazioni Java Tomcat con un front-end Apache.
- Tomcat è configurato per consentire alle applicazioni di utilizzare un `DataSource` oggetto JDBC per connettersi a un'istanza MySQL separata, che funge da archivio dati di backend.

Il libro di ricette per questo progetto coinvolge diversi componenti chiave:

- Il file di attributi contiene impostazioni di configurazione utilizzate dalle diverse ricette.

- Le ricette di impostazione sono assegnate all'evento del ciclo di vita Setup del livello. Vengono eseguite dopo l'avvio di un'istanza ed eseguono attività quali l'installazione di pacchetti e la creazione di file di configurazione.
- Le ricette di configurazione sono assegnate all'evento del ciclo di vita Configure del livello. Vengono eseguiti dopo le modifiche alla configurazione dello stack, principalmente quando le istanze sono online o offline, e gestiscono tutte le modifiche di configurazione richieste.
- Le ricette di ditribuzione sono assegnate all'evento del ciclo di vita Deploy del livello. Vengono eseguite dopo le ricette di impostazione e quando distribuisci manualmente un'app per installare il codice e i file correlati nelle istanze di un livello e per gestire le attività correlate, come il riavvio dei servizi.

La sezione finale descrive come creare uno stack che includa un livello personalizzato basato sul ricettario Tomcat e come distribuire ed eseguire una semplice applicazione JSP che visualizza i dati da un database MySQL in esecuzione su un'istanza che appartiene a un livello MySQL separato. Creare uno stack ed eseguire un'applicazione

> **Note**
>
> Le ricette del ricettario Tomcat dipendono da alcune ricette integrate in Stacks. AWS OpsWorks Affinché l'origine di ogni ricetta sia chiara, questo argomento identifica ricette utilizzando la convenzione di Chef 1cookbookname::recipename.

Argomenti

- File di attributi
- Ricette di impostazione
- Ricette di configurazione
- Ricette di ditribuzione
- Creare uno stack ed eseguire un'applicazione

File di attributi

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Prima di approfondire le ricette, è utile esaminare il file di attributi del libro di ricette Tomcat, che contiene diverse impostazioni di configurazione utilizzate dalle ricette. Gli attributi non sono necessari. Puoi semplicemente impostare come hardcoded tali valori nelle tue ricette o nei tuoi modelli. Tuttavia, se definisci le impostazioni di configurazione utilizzando gli attributi, puoi utilizzare la console o l'API AWS OpsWorks Stacks per modificare i valori definendo attributi JSON personalizzati, un'operazione più semplice e flessibile rispetto alla riscrittura della ricetta o del codice del modello ogni volta che desideri modificare un'impostazione. Tale approccio ti consente, ad esempio, di utilizzare lo stesso libro di ricette per più stack, ma presuppone una diversa configurazione del server Tomcat per ogni stack. Per ulteriori informazioni sugli attributi e su come sostituirli, consulta Sostituzione degli attributi.

L'esempio seguente mostra il file degli attributi completo `default.rb`, presente nella directory `attributes` del libro di ricette Tomcat.

```
default['tomcat']['base_version'] = 6
default['tomcat']['port'] = 8080
default['tomcat']['secure_port'] = 8443
default['tomcat']['ajp_port'] = 8009
default['tomcat']['shutdown_port'] = 8005
default['tomcat']['uri_encoding'] = 'UTF-8'
default['tomcat']['unpack_wars'] = true
default['tomcat']['auto_deploy'] = true
case node[:platform]
when 'centos', 'redhat', 'fedora', 'amazon'
  default['tomcat']['java_opts'] = ''
when 'debian', 'ubuntu'
  default['tomcat']['java_opts'] = '-Djava.awt.headless=true -Xmx128m -XX:
+UseConcMarkSweepGC'
end
default['tomcat']['catalina_base_dir'] = "/etc/tomcat#{node['tomcat']['base_version']}"
default['tomcat']['webapps_base_dir'] = "/var/lib/tomcat#{node['tomcat']
['base_version']}/webapps"
default['tomcat']['lib_dir'] = "/usr/share/tomcat#{node['tomcat']['base_version']}/lib"
default['tomcat']['java_dir'] = '/usr/share/java'
default['tomcat']['mysql_connector_jar'] = 'mysql-connector-java.jar'
```

```
default['tomcat']['apache_tomcat_bind_mod'] = 'proxy_http' # or: 'proxy_ajp'
default['tomcat']['apache_tomcat_bind_config'] = 'tomcat_bind.conf'
default['tomcat']['apache_tomcat_bind_path'] = '/tc/'
default['tomcat']['webapps_dir_entries_to_delete'] = %w(config log public tmp)
case node[:platform]
when 'centos', 'redhat', 'fedora', 'amazon'
  default['tomcat']['user'] = 'tomcat'
  default['tomcat']['group'] = 'tomcat'
  default['tomcat']['system_env_dir'] = '/etc/sysconfig'
when 'debian', 'ubuntu'
  default['tomcat']['user'] = "tomcat#{node['tomcat']['base_version']}"
  default['tomcat']['group'] = "tomcat#{node['tomcat']['base_version']}"
  default['tomcat']['system_env_dir'] = '/etc/default'
end
```

Le impostazioni vengono descritte più avanti nella sezione correlata. In generale, si applicano le note seguenti:

- Poiché tutte le definizioni di nodo sono del tipo `default`, puoi sostituirle con gli attributi JSON personalizzati.
- Il file utilizza un'istruzione `case` per impostare condizionalmente alcuni valori di attributi in base al sistema operativo dell'istanza.

  Il nodo `platform` è generato dallo strumento Ohai di Chef e rappresenta il sistema operativo dell'istanza.

Ricette di impostazione

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le ricette di impostazione sono assegnate all'evento del ciclo di vita Setup del livello e vengono eseguite dopo l'avvio di un'istanza. Eseguono attività quali l'installazione di pacchetti, la creazione di

file di configurazione e l'avvio di servizi. Al termine dell'esecuzione delle ricette di installazione, AWS OpsWorks Stacks esegue le ricette Deploy per distribuire qualsiasi app nella nuova istanza.

Argomenti

- tomcat::setup
- tomcat::install
- tomcat::service
- tomcat::container_config
- tomcat::apache_tomcat_bind

tomcat::setup

La ricetta `tomcat::setup` è stata concepita per essere assegnata a un evento del ciclo di vita Setup del livello.

```
include_recipe 'tomcat::install'
include_recipe 'tomcat::service'

service 'tomcat' do
  action :enable
end

# for EBS-backed instances we rely on autofs
bash '(re-)start autofs earlier' do
  user 'root'
  code <<-EOC
    service autofs restart
  EOC
  notifies :restart, resources(:service => 'tomcat')
end

include_recipe 'tomcat::container_config'
include_recipe 'apache2'
include_recipe 'tomcat::apache_tomcat_bind'
```

La ricetta `tomcat::setup` è in gran parte una meta-ricetta. Include un set di ricette dipendenti che gestiscono la maggior parte dei dettagli di installazione e configurazione di Tomcat e dei pacchetti correlati. La prima parte di `tomcat::setup` esegue le seguenti ricette, descritte più avanti:

- La ricetta tomcat::install installa il pacchetto del server Tomcat.
- La ricetta tomcat::service configura il servizio Tomcat.

La parte centrale di `tomcat::setup` abilita e avvia il servizio Tomcat:

- La risorsa del servizio Chef abilita il servizio Tomcat all'avvio.
- La risorsa Chef bash esegue uno script Bash per avviare il demone autofs, necessario per le istanze supportate da Amazon EBS. La risorsa, quindi, comunica alla risorsa `service` di riavviare il servizio Tomcat.

  Per ulteriori informazioni, consulta: autofs (per Amazon Linux) o Autofs (per Ubuntu).

La parte finale di `tomcat::setup` crea i file di configurazione e installa e configura il server front-end Apache:

- La ricetta tomcat::container_config crea file di configurazione.
- La `apache2` ricetta (abbreviazione di) è una ricetta integrata di Stacks che installa e configura un `apache2::default` AWS OpsWorks server Apache.
- La ricetta tomcat::apache_tomcat_bind configura il server Apache affinché funga da front-end per il server Tomcat.

> **Note**
>
> Spesso puoi risparmiare tempo e risorse utilizzando ricette integrate per eseguire alcune delle attività necessarie. Questa ricetta utilizza la ricetta `apache2::default` integrata per installare Apache anziché implementarlo da zero. Per un altro esempio di come utilizzare ricette integrate, consulta Ricette di ditribuzione.

Nelle seguenti sezioni vengono descritte nel dettaglio le ricette di impostazione del libro di ricette Tomcat. Per ulteriori informazioni sulle ricette `apache2`, consulta opsworks-cookbooks/apache2.

tomcat::install

La ricetta `tomcat::install` installa il server Tomcat, OpenJDK e una libreria di connettori Java che gestisce la connessione al server MySQL.

```
tomcat_pkgs = value_for_platform(
  ['debian', 'ubuntu'] => {
    'default' => ["tomcat#{node['tomcat']['base_version']}", 'libtcnative-1',
 'libmysql-java']
  },
  ['centos', 'redhat', 'fedora', 'amazon'] => {
    'default' => ["tomcat#{node['tomcat']['base_version']}", 'tomcat-native', 'mysql-
connector-java']
  },
  'default' => ["tomcat#{node['tomcat']['base_version']}"]
)

tomcat_pkgs.each do |pkg|
  package pkg do
    action :install
  end
end

link ::File.join(node['tomcat']['lib_dir'], node['tomcat']['mysql_connector_jar']) do
  to ::File.join(node['tomcat']['java_dir'], node['tomcat']['mysql_connector_jar'])
  action :create
end

# remove the ROOT webapp, if it got installed by default
include_recipe 'tomcat::remove_root_webapp'
```

La ricetta esegue le seguenti attività:

1. Crea un elenco di pacchetti da installare, a seconda del sistema operativo dell'istanza.
2. Installa ogni pacchetto nell'elenco.

   La risorsa del pacchetto Chef utilizza il provider appropriato, `yum` per Amazon Linux e `apt-get` Ubuntu, per gestire l'installazione. I provider di pacchetti installano OpenJDK come dipendenza Tomcat, ma la libreria di connettori MySQL deve essere installata in modo esplicito.
3. Utilizza una risorsa link di Chef per creare un collegamento simbolico nella directory lib del server Tomcat alla libreria dei connettori MySQL in JDK.

   In base ai valori degli attributi predefiniti, la directory lib di Tomcat è `/usr/share/tomcat6/lib` e la libreria dei connettori MySQL (`mysql-connector-java.jar`) è in `/usr/share/java/`.

La ricetta `tomcat::remove_root_webapp` rimuove l'applicazione Web ROOT (`/var/lib/tomcat6/webapps/ROOT` per impostazione predefinita) per evitare alcune problematiche relative alla sicurezza.

```
ruby_block 'remove the ROOT webapp' do
  block do
    ::FileUtils.rm_rf(::File.join(node['tomcat']['webapps_base_dir'], 'ROOT'), :secure
 => true)
  end
  only_if { ::File.exists?(::File.join(node['tomcat']['webapps_base_dir'], 'ROOT'))
 && !::File.symlink?(::File.join(node['tomcat']['webapps_base_dir'], 'ROOT')) }
end
```

L'istruzione `only_if` garantisce che la ricetta rimuoverà il file solo se esistente.

> **Note**
>
> La versione di Tomcat viene specificata dall'attributo `['tomcat']['base_version']`, impostato su 6 nel file degli attributi. Per installare Tomcat 7, puoi sostituire l'attributo con gli attributi JSON personalizzati. È sufficiente che modifichi le impostazioni dello stack e inserisci il seguente JSON nella casella Custom Chef JSON (JSON Chef personalizzato) oppure lo aggiungi a un JSON personalizzato esistente:
>
> ```
> {
>   'tomcat' : {
>     'base_version' : 7
>   }
> }
> ```
>
> L'attributo JSON personalizzato sostituisce l'attributo predefinito e imposta la versione di Tomcat su 7. Per ulteriori informazioni sulla sostituzione di attributi, consulta Sostituzione degli attributi.

tomcat::service

La ricetta `tomcat::service` crea la definizione del servizio Tomcat.

```
service 'tomcat' do
  service_name "tomcat#{node['tomcat']['base_version']}"

  case node[:platform]
  when 'centos', 'redhat', 'fedora', 'amazon'
    supports :restart => true, :reload => true, :status => true
  when 'debian', 'ubuntu'
    supports :restart => true, :reload => false, :status => true
  end

  action :nothing
end
```

La ricetta usa la risorsa service di Chef per specificare il nome di servizio Tomcat (tomcat6, per impostazione predefinita) e imposta l'attributo `supports` per definire in che modo Chef gestisce il riavvio, il ricaricamento e i comandi di stato del servizio sui vari sistemi operativi.

- `true` indica che Chef può utilizzare lo script init o un altro provider di servizi per eseguire il comando
- `false` indica che Chef deve tentare di eseguire il comando con altri mezzi.

`action` è impostato su `:nothing`. Per ogni evento del ciclo di vita, AWS OpsWorks Stacks avvia un'esecuzione Chef per eseguire il set di ricette appropriato. Il libro di ricette Tomcat segue un modello comune in base al quale una ricetta crea la definizione del servizio, ma non lo riavvia. Altre ricette nell'esecuzione Chef gestiscono il riavvio, di solito includendo un comando `notifies` nelle risorse `template` utilizzate per creare i file di configurazione. Le notifiche sono utili per riavviare un servizio poiché lo fanno solo se la configurazione è cambiata. Inoltre, se un'esecuzione Chef include più notifiche di riavvio per un servizio, Chef riavvia il servizio non più di una volta. Questa pratica evita i problemi che possono verificarsi durante il tentativo di riavvio di un servizio non completamente operativo, fonte comune di errori Tomcat.

Il servizio Tomcat deve essere definito per qualsiasi esecuzione di Chef che utilizza le notifiche di riavvio. `tomcat::service` pertanto è incluso in diverse ricette, per assicurare che il servizio sia definito per ogni esecuzione di Chef. Non vi è alcuna sanzione se un'esecuzione Chef include più istanze di `tomcat::service` perché Chef garantisce una singola esecuzione per ricetta, indipendentemente dal numero di volte che tale ricetta viene inclusa.

tomcat::container_config

La ricetta `tomcat::container_config` crea file di configurazione dai file modello del libro di ricette.

```
include_recipe 'tomcat::service'

template 'tomcat environment configuration' do
  path ::File.join(node['tomcat']['system_env_dir'], "tomcat#{node['tomcat']
['base_version']}")
  source 'tomcat_env_config.erb'
  owner 'root'
  group 'root'
  mode 0644
  backup false
  notifies :restart, resources(:service => 'tomcat')
end

template 'tomcat server configuration' do
  path ::File.join(node['tomcat']['catalina_base_dir'], 'server.xml')
  source 'server.xml.erb'
  owner 'root'
  group 'root'
  mode 0644
  backup false
  notifies :restart, resources(:service => 'tomcat')
end
```

Innanzitutto la ricetta chiama `tomcat::service`, che definisce il servizio, se necessario. Il blocco principale della ricetta consiste di due risorse template, ciascuna delle quali crea un file di configurazione da uno dei file modello del libro di ricette, imposta le proprietà del file e comunica a Chef di riavviare il servizio.

File di configurazione dell'ambiente Tomcat

La prima risorsa `template` usa il file modello `tomcat_env_config.erb` per creare un file di configurazione dell'ambiente Tomcat, utilizzato per impostare variabili di ambiente, come `JAVA_HOME`. Il nome file predefinito è l'argomento della risorsa `template`. `tomcat::container_config` utilizza un attributo `path` per sostituire il valore predefinito e nominare il file di configurazione `/etc/sysconfig/tomcat6` (Amazon Linux) o `/etc/default/`

`tomcat6` (Ubuntu). La risorsa `template` specifica inoltre il proprietario del file, il gruppo e la modalità e indica a Chef di non creare file di backup.

Se esamini il codice sorgente, vedrai che esistono di fatto tre versioni di `tomcat_env_config.erb`, ognuna in una sottodirectory diversa della directory `templates`. Le directory `ubuntu` e `amazon` contengono i modelli per i rispettivi sistemi operativi. La cartella `default` contiene un modello fittizio con una singola riga di commento, che viene utilizzato solo se cerchi di eseguire questa ricetta su un'istanza con un sistema operativo non supportato. Per la ricetta `tomcat::container_config` non è necessario specificare quale `tomcat_env_config.erb` utilizzare. Chef seleziona automaticamente la directory appropriata al sistema operativo dell'istanza in funzione delle regole descritte in Specificità file.

I file `tomcat_env_config.erb` di questo esempio consistono in gran parte di commenti. Per impostare ulteriori variabili di ambiente, annulla semplicemente i commenti delle righe appropriate e fornisci i valori che preferisci.

> **Note**
>
> Qualsiasi impostazione di configurazione che potrebbe cambiare deve essere definita come attributo anziché come impostazione hardcoded nel modello. In questo modo, non dovrai riscrivere il modello per modificare un'impostazione, ma semplicemente sostituire l'attributo.

Il modello di Amazon Linux imposta solo una variabile di ambiente, come illustrato nel seguente estratto.

```
...
# Use JAVA_OPTS to set java.library.path for libtcnative.so
#JAVA_OPTS="-Djava.library.path=/usr/lib"

JAVA_OPTS="${JAVA_OPTS} <%= node['tomcat']['java_opts'] %>"

# What user should run tomcat
#TOMCAT_USER="tomcat"
...
```

JAVA_OPTS può essere utilizzato per specificare opzioni Java, come il percorso della libreria. Utilizzando valori di attributi predefiniti, il modello non imposta alcuna opzione Java per Amazon

Linux. Puoi impostare le tue opzioni Java sostituendo l'attributo `['tomcat']['java_opts']`, ad esempio, con gli attributi JSON personalizzati. Per vedere un esempio, consulta Creare uno stack.

Il modello di Ubuntu imposta diverse variabili di ambiente, come illustrato nel seguente estratto del modello.

```
# Run Tomcat as this user ID. Not setting this or leaving it blank will use the
# default of tomcat<%= node['tomcat']['base_version'] %>.
TOMCAT<%= node['tomcat']['base_version'] %>_USER=tomcat<%= node['tomcat']
['base_version'] %>
...
# Run Tomcat as this group ID. Not setting this or leaving it blank will use
# the default of tomcat<%= node['tomcat']['base_version'] %>.
TOMCAT<%= node['tomcat']['base_version'] %>_GROUP=tomcat<%= node['tomcat']
['base_version'] %>
...
JAVA_OPTS="<%= node['tomcat']['java_opts'] %>"

<% if node['tomcat']['base_version'].to_i < 7 -%>
# Unset LC_ALL to prevent user environment executing the init script from
# influencing servlet behavior.  See Debian bug #645221
unset LC_ALL
<% end -%>
```

Utilizzando valori di attributi predefiniti, il modello imposta le variabili dell'ambiente Ubuntu come segue:

- `TOMCAT6_USER` e `TOMCAT6_GROUP`, che rappresentano l'utente e il gruppo di Tomcat, sono entrambe impostate su `tomcat6`.

  Se imposti ['tomcat']['base_version'] su `tomcat7`, i nomi delle variabili si risolvono in `TOMCAT7_USER` e `TOMCAT7_GROUP` ed entrambi sono impostati su `tomcat7`.
- `JAVA_OPTS` è impostato su `-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`:
  - L'impostazione di `-Djava.awt.headless` su `true` informa il motore di grafica che l'istanza è headless e non dispone di una console, che risolve il comportamento errato di alcune applicazioni grafiche.
  - `-Xmx128m` garantisce che JVM dispone di adeguate risorse di memoria, in questo esempio 128 MB.

- `-XX:+UseConcMarkSweepGC` specifica la garbage collection Concurrent Mark Sweep (CMS), che consente di limitare le pause indotte da garbage collection.

  Per ulteriori informazioni, consulta Concurrent Mark Sweep Collector Enhancements.
- Se la versione di Tomcat è precedente alla 7, il modello annulla l'impostazione `LC_ALL`, che risolve un bug di Ubuntu.

> **Note**
>
> Con gli attributi predefiniti, alcune di queste variabili di ambiente sono semplicemente impostate sui valori predefiniti. Tuttavia, l'impostazione esplicita delle variabili di ambiente sugli attributi indica che puoi definire attributi JSON personalizzati con cui sostituire gli attributi predefiniti e fornire valori personalizzati. Per ulteriori informazioni sulla sostituzione di attributi, consulta Sostituzione degli attributi.

Per completare i file modello, consulta il codice sorgente.

File di configurazione Server.xml

La seconda `template` risorsa viene utilizzata `server.xml.erb` per creare il file di `system.xml` configurazione, che configura il contenitore. servlet/JSP `server.xml.erb`non contiene impostazioni specifiche del sistema operativo, quindi si trova nella `template` sottodirectory della directory. `default`

Il modello utilizza le impostazioni standard, ma può creare un file `system.xml` per Tomcat 6 o Tomcat 7. Ad esempio, il codice seguente della sezione server del modello configura i listener in modo appropriato per la versione specificata.

```
<% if node['tomcat']['base_version'].to_i > 6 -%>
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener" />
  -->
<% end -%>
  <!--APR library loader. Documentation at /docs/apr.html -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <!--Initialize Jasper prior to webapps are loaded. Documentation at /docs/jasper-
howto.html -->
```

```
  <Listener className="org.apache.catalina.core.JasperListener" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
<% if node['tomcat']['base_version'].to_i < 7 -%>
  <!-- JMX Support for the Tomcat server. Documentation at /docs/non-existent.html -->
  <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener" />
<% end -%>
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
<% if node['tomcat']['base_version'].to_i > 6 -%>
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
<% end -%>
```

Il modello utilizza gli attributi al posto delle impostazioni hardcoded affinché tu possa facilmente modificare le impostazioni mediante la definizione di attributi JSON personalizzati. Per esempio:

```
<Connector port="<%= node['tomcat']['port'] %>" protocol="HTTP/1.1"
           connectionTimeout="20000"
           URIEncoding="<%= node['tomcat']['uri_encoding'] %>"
           redirectPort="<%= node['tomcat']['secure_port'] %>" />
```

Per ulteriori informazioni, consulta il codice sorgente.

tomcat::apache_tomcat_bind

La ricetta `tomcat::apache_tomcat_bind` consente al server Apache di funzionare come front-end di Tomcat, in quanto riceve le richieste in arrivo e le inoltra a Tomcat per poi inviare le risposte al client. Questo esempio utilizza mod_proxy come proxy/gateway Apache.

```
execute 'enable mod_proxy for apache-tomcat binding' do
  command '/usr/sbin/a2enmod proxy'
  not_if do
    ::File.symlink?(::File.join(node['apache']['dir'], 'mods-enabled', 'proxy.load'))
 || node['tomcat']['apache_tomcat_bind_mod'] !~ /\Aproxy/
  end
end

execute 'enable module for apache-tomcat binding' do
  command "/usr/sbin/a2enmod #{node['tomcat']['apache_tomcat_bind_mod']}"
  not_if {::File.symlink?(::File.join(node['apache']['dir'], 'mods-enabled',
 "#{node['tomcat']['apache_tomcat_bind_mod']}.load"))}
end
```

```
include_recipe 'apache2::service'

template 'tomcat thru apache binding' do
  path ::File.join(node['apache']['dir'], 'conf.d', node['tomcat']
['apache_tomcat_bind_config'])
  source 'apache_tomcat_bind.conf.erb'
  owner 'root'
  group 'root'
  mode 0644
  backup false
  notifies :restart, resources(:service => 'apache2')
end
```

Per abilitare mod_proxy, devi abilitare il modulo `proxy` e un modulo basato sul protocollo. Per il modulo del protocollo sono disponibili due opzioni:

- HTTP: `proxy_http`
- JServ Protocollo Apache (AJP): `proxy_ajp`

  AJP è un protocollo di Tomcat interno.

Entrambe le risorse execute della ricetta eseguono il comando `a2enmod`, che abilita il modulo specificato mediante la creazione dei collegamenti simbolici richiesti:

- La prima risorsa `execute` abilita il modulo `proxy`.
- La seconda risorsa `execute` abilita il modulo del protocollo, che è configurato su `proxy_http` per impostazione predefinita.

  Se preferisci utilizzare AJP, puoi definire un JSON personalizzato per sostituire l'attributo `apache_tomcat_bind_mod` e impostarlo su `proxy_ajp`.

La `apache2::service` ricetta è una ricetta integrata di AWS OpsWorks Stacks che definisce il servizio Apache. Per ulteriori informazioni, consulta la ricetta nel repository AWS OpsWorks GitHub Stacks.

La risorsa `template` utilizza `apache_tomcat_bind.conf.erb` per creare un file di configurazione, denominato `tomcat_bind.conf` per impostazione predefinita. Colloca il file nella directory `['apache']['dir']/.conf.d`. L'attributo `['apache']['dir']` viene definito nel file

di attributi `apache2` integrato ed è configurato su `/etc/httpd` (Amazon Linux) oppure su `/etc/apache2` (Ubuntu) per impostazione predefinita. Se la risorsa `template` crea o modifica il file di configurazione, il comando `notifies` pianifica un riavvio del servizio Apache.

```
<% if node['tomcat']['apache_tomcat_bind_mod'] == 'proxy_ajp' -%>
ProxyPass <%= node['tomcat']['apache_tomcat_bind_path'] %> ajp://localhost:<%=
 node['tomcat']['ajp_port'] %>/
ProxyPassReverse <%= node['tomcat']['apache_tomcat_bind_path'] %> ajp://localhost:<%=
 node['tomcat']['ajp_port'] %>/
<% else %>
ProxyPass <%= node['tomcat']['apache_tomcat_bind_path'] %> http://localhost:<%=
 node['tomcat']['port'] %>/
ProxyPassReverse <%= node['tomcat']['apache_tomcat_bind_path'] %> http://localhost:<%=
 node['tomcat']['port'] %>/
<% end -%>
```

Il modello utilizza le ProxyPassReversedirettive ProxyPassand per configurare la porta utilizzata per far passare il traffico tra Apache e Tomcat. Poiché sono sulla stessa istanza, entrambi i server possono utilizzare un URL localhost ed essere configurati su `http://localhost:8080` per impostazione predefinita.

Ricette di configurazione

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le ricette di configurazione sono assegnate all'evento del ciclo di vita Configure del livello, che si verifica su tutte le istanze dello stack ogni volta che un'istanza entra o esce dallo stato online. Puoi utilizzare le ricette di configurazione per regolare la configurazione di un'istanza in modo da rispondere alla modifica in maniera appropriata. Quando implementi una ricetta Configure, tieni presente che la modifica della configurazione di uno stack potrebbe coinvolgere istanze che non hanno nulla a che fare con questo livello. La ricetta deve essere in grado di rispondere in modo appropriato, che in alcuni casi potrebbe significare non fare nulla.

tomcat::configure

La ricetta `tomcat::configure` è destinata a un evento del ciclo di vita Configure del livello.

```
include_recipe 'tomcat::context'
# Optional: Trigger a Tomcat restart in case of a configure event, if relevant
# settings in custom JSON have changed (e.g. java_opts/JAVA_OPTS):
#include_recipe 'tomcat::container_config'
```

La ricetta `tomcat::configure` è fondamentalmente un meta-ricetta che esegue due ricette dipendenti.

1. La ricetta `tomcat::context` crea un file di configurazione del contesto dell'app Web.

   Questo file configura le risorse JDBC che le applicazioni utilizzano per comunicare con l'istanza MySQL, come illustrato nella sezione successiva. Eseguire questa ricetta in risposta a un evento di configurazione consente al livello di aggiornare il file di configurazione del contesto dell'app Web se il livello del database è stato modificato.
2. La ricetta Setup `tomcat::container_config` viene eseguita nuovamente per acquisire eventuali modifiche apportate alla configurazione del container.

In questo esempio viene commentato `include` per `tomcat::container_config`. Se desideri utilizzare un JSON personalizzato per modificare le impostazioni di Tomcat, puoi rimuovere il commento. Un evento del ciclo di vita Configure esegue quindi `tomcat::container_config`, che aggiorna i file di configurazione correlati a Tomcat, come descritto in tomcat::container_config e riavvia il servizio Tomcat.

tomcat::context

Il ricettario Tomcat consente alle applicazioni di accedere a un server di database MySQL, che può essere eseguito su un'istanza separata, utilizzando un oggetto J2EE. DataSource Con Tomcat, puoi abilitare la connessione mediante la creazione e l'installazione di un file di configurazione del contesto dell'app Web per ogni applicazione. Questo file definisce il rapporto tra l'applicazione e la risorsa JDBC che verrà utilizzata dall'applicazione per comunicare con il database. Per maggiori informazioni, consulta The Context Container.

Lo scopo primario della ricetta `tomcat::context` è creare questo file di configurazione.

```
include_recipe 'tomcat::service'

node[:deploy].each do |application, deploy|
  context_name = deploy[:document_root].blank? ? application : deploy[:document_root]

  template "context file for #{application} (context name: #{context_name})" do
    path ::File.join(node['tomcat']['catalina_base_dir'], 'Catalina', 'localhost',
 "#{context_name}.xml")
    source 'webapp_context.xml.erb'
    owner node['tomcat']['user']
    group node['tomcat']['group']
    mode 0640
    backup false
    only_if { node['datasources'][context_name] }
    variables(:resource_name => node['datasources'][context_name], :webapp_name =>
 application)
    notifies :restart, resources(:service => 'tomcat')
  end
end
```

Oltre agli attributi del libro di cucina Tomcat, questa ricetta utilizza gli attributi di configurazione e distribuzione dello stack che Stacks installa con l'evento Configure. AWS OpsWorks Il servizio AWS OpsWorks Stacks aggiunge attributi all'oggetto nodo di ogni istanza che contengono le informazioni che le ricette in genere otterrebbero utilizzando i data bag o la ricerca e installa gli attributi su ogni istanza. Gli attributi contengono informazioni dettagliate sulla configurazione dello stack, sulle app distribuite e sui dati personalizzati che un utente desidera includere. Le ricette possono ottenere dati da attributi di configurazione e di distribuzione dello stack utilizzando una sintassi del nodo Chef standard. Per ulteriori informazioni, consulta Attributi di configurazione e distribuzione dello stack. Con gli stack Chef 11.10, puoi usare la ricerca Chef per ottenere i dati di configurazione e distribuzione dello stack. Per ulteriori informazioni, consulta Utilizzo della ricerca di Chef.

`deploy`attributes si riferisce allo spazio dei `[:deploy]` nomi, che contiene gli attributi relativi alla distribuzione definiti tramite la console o l'API o generati dal servizio Stacks. AWS OpsWorks L'attributo `deploy` include un attributo per ogni app distribuita, denominata con il nome breve dell'app. Ogni attributo di app contiene un set di attributi che caratterizzano l'app, ad esempio la radice del documento (`[:deploy][:`*`appname`*`][:document_root]`).

La ricetta `context` innanzitutto assicura che il servizio viene definito per questa esecuzione Chef chiamando tomcat::service. Quindi definisce una variabile `context_name` che rappresenta il nome del file di configurazione, esclusa l'estensione `.xml`. Se utilizzi la radice del documento predefinita, `context_name` viene impostato sul nome breve dell'app. In alternativa, viene impostato sulla radice

del documento specificato. L'esempio illustrato in Creare uno stack ed eseguire un'applicazione imposta la radice del documento su `"ROOT"`, pertanto il contesto è ROOT e il file di configurazione viene denominato `ROOT.xml`.

Il blocco principale della ricetta scorre l'elenco delle app distribuite e, per ogni app, usa il modello `webapp_context.xml.erb` per creare un file di configurazione del contesto. L'esempio distribuisce solo un'app, ma la definizione dell'attributo `deploy` richiede di trattarla come un elenco di applicazioni.

Il modello `webapp_context.xml.erb` non è specifico del sistema operativo, pertanto si trova nella sottodirectory `templates` della directory `default`.

La ricetta crea il file di configurazione come segue:

- Grazie a valori di attributo predefiniti, il nome del file di configurazione viene impostato su *`context_name`*`.xml` e installato nella directory `/etc/tomcat6/Catalina/localhost/`.

  Il nodo `['datasources']` degli attributi di configurazione dello stack contiene uno o più attributi, ciascuno dei quali mappa un nome di contesto alla risorsa di dati JDBC che l'applicazione associata utilizzerà per comunicare con il database. Il nodo e i relativi contenuti vengono definiti con un JSON personalizzato al momento della creazione dello stack, come descritto di seguito in Creare uno stack ed eseguire un'applicazione. L'esempio dispone di un singolo attributo che associa il nome di contesto ROOT a una risorsa JDBC denominata jdbc/mydb.
- Mediante l'utilizzo di valori di attributo predefiniti, il gruppo e l'utente del file sono entrambi impostati sui valori definiti dal pacchetto Tomcat: `tomcat` (Amazon Linux) o `tomcat6` (Ubuntu).
- La risorsa `template` crea il file di configurazione solo se il nodo `['datasources']` esiste e include un attributo `context_name`.
- La risorsa `template` definisce due variabili: `resource_name` e `webapp_name`.

  `resource_name` è impostato sul nome di risorsa associato a `context_name` mentre `webapp_name` è impostato sul nome breve dell'app.
- La risorsa modello riavvia il servizio Tomcat per caricare e attivare le modifiche.

Il modello `webapp_context.xml.erb` è costituito da un elemento `Context` che contiene un elemento `Resource` con il relativo set di attributi.

Gli attributi caratterizzano la configurazione del contesto: `Resource`

- name: il nome della risorsa JDBC, impostato sul `resource_name` valore definito in. `tomcat::context`

  Ad esempio, il nome della risorsa è impostato su jdbc/mydb.
- auth e type: si tratta di impostazioni standard per le connessioni JDBC. `DataSource`
- MaxActive, MaxIdle e MaxWait: il numero massimo di connessioni attive e inattive e il tempo di attesa massimo per la restituzione di una connessione.
- nome utente e password: nome utente e password root del database, ottenuti dagli attributi. `deploy`
- driverClassName—Il nome della classe del driver JDBC, impostato sul driver MySQL.
- url: l'URL di connessione.

  Il prefisso dipende dal database. Deve essere impostato su `jdbc:mysql` per MySQL, su `jdbc:postgresql` per Postgres e su `jdbc:sqlserver` per SQL Server. L'esempio imposta l'URL su`jdbc:mysql://`*`host_IP_Address`*`:3306:simplejsp`, where *`simplejsp`* è il nome breve dell'app.
- factory: la `DataSource` fabbrica, necessaria per i database MySQL.

Per ulteriori informazioni su questo file di configurazione, consulta l'argomento Utilizzo del wiki di Tomcat. DataSources

Ricette di ditribuzione

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le ricette di ditribuzione sono assegnate all'evento del ciclo di vita Deploy del livello. In genere si verifica su tutte le istanze dello stack ogni volta che si distribuisce un'app, sebbene sia possibile facoltativamente limitare l'evento solo a istanze specifiche. AWS OpsWorks Stacks esegue anche le ricette Deploy su nuove istanze, dopo il completamento delle ricette di installazione. Lo scopo

primario delle ricette di ditribuzione è distribuire codice e file correlati da un repository alle istanze del livello del server di applicazioni. Tuttavia, esegui spesso ricette di ditribuzione anche su altri livelli. Questo consente alle istanze di tali livelli, ad esempio, di aggiornare la propria configurazione per includere l'app appena distribuita. Quando implementi una ricetta Deploy, ricorda che un evento Deploy non significa necessariamente che le app vengono distribuite nell'istanza. Potrebbe trattarsi semplicemente di una notifica del fatto che le app vengono distribuite in altre istanze dello stack per consentire all'istanza di effettuare gli aggiornamenti necessari. La ricetta deve essere in grado di rispondere in modo appropriato, che potrebbe significare anche non fare nulla.

AWS OpsWorks Stacks distribuisce automaticamente le app dei tipi di app standard nei corrispondenti livelli di server di applicazioni integrati. Per distribuire app in un livello personalizzato, devi implementare ricette di ditribuzione personalizzate che eseguono il download dei file dell'app da un repository nel percorso appropriato sull'istanza. Tuttavia, spesso puoi limitare la quantità di codice che devi scrivere utilizzando il libro di ricette di ditribuzione integrato per gestire alcuni aspetti della distribuzione. Ad esempio, se archivi i tuoi file in uno dei repository supportati, il libro di ricette integrato può gestire i dettagli del download dei file dal repository nelle istanze del livello.

La ricetta `tomcat::deploy` è concepita per essere assegnata all'evento del ciclo di vita Deploy.

```
include_recipe 'deploy'

node[:deploy].each do |application, deploy|
  opsworks_deploy_dir do
    user deploy[:user]
    group deploy[:group]
    path deploy[:deploy_to]
  end

  opsworks_deploy do
    deploy_data deploy
    app application
  end
...
```

La ricetta `tomcat::deploy` utilizza il libro di ricette di ditribuzione integrato per aspetti della distribuzione che non sono specifici dell'applicazione. La ricetta `deploy` (forma abbreviata per la ricetta `deploy::default` integrata) è una ricetta integrata che gestisce i dettagli di configurazione di utenti, gruppi e così via, in base ai dati degli attributi `deploy`.

La ricetta usa due definizioni Chef integrate, `opsworks_deploy_dir` e opworks_deploy, per installare l'applicazione.

La definizione `opsworks_deploy_dir` configura la struttura della directory in base ai dati del JSON di distribuzione dell'app. Le definizioni sono molto utili per creare un pacchetto con le definizioni delle risorse e si trovano nella directory `definitions` di un libro di ricette. Le ricette possono utilizzare le definizioni in modo analogo alle risorse, tuttavia la definizione stessa non ha un provider associato, ma solo le risorse incluse nella definizione. Puoi definire variabili nella ricetta, che vengono passate alle definizioni di risorse sottostanti. La ricetta `tomcat::deploy` imposta le variabili `user`, `group` e `path` in base ai dati del JSON di distribuzione. Questi vengono passati alla risorsa di directory della definizione, che gestisce le directory.

> **Note**
>
> L'utente e il gruppo dell'app distribuita sono determinati dagli `[:opsworks][:deploy_user][:group]` attributi `[:opsworks][:deploy_user][:user]` e, definiti nel file degli attributi del cookbook di deploy integrato. `deploy.rb` Il valore predefinito di `[:opsworks][:deploy_user][:user]` è `deploy`. Il valore predefinito di `[:opsworks][:deploy_user][:group]` dipende dal sistema operativo dell'istanza:
>
> - Per le istanze Ubuntu, il gruppo predefinito è `www-data`.
> - Per le istanze Amazon Linux che sono membri di un livello Rails App Server che utilizza Nginx e Unicorn, il gruppo predefinito è. `nginx`
> - Per tutte le altre istanze Amazon Linux, il gruppo predefinito è `apache`.
>
> Puoi modificare le impostazioni utilizzando un JSON personalizzato o un file di attributi personalizzato per sostituire l'attributo appropriato. Per ulteriori informazioni, consulta Sostituzione degli attributi.

L'altra definizione, `opsworks_deploy`, gestisce i dettagli di verifica del codice dell'app e dei file correlati del repository e della loro distribuzione nell'istanza, in base ai dati degli attributi `deploy`. Puoi utilizzare questa definizione per qualsiasi tipo di app. I dettagli di distribuzione, come i nomi di directory, sono specificati nella console o tramite l'API e inseriti negli attributi `deploy`. Tuttavia, `opsworks_deploy` funziona solo per i quattro tipi di repository supportati: Git, Subversion, S3 e HTTP. Se desideri utilizzare un altro tipo di repository, devi implementare personalmente questo codice.

Installi file di un'app nella directory `webapps` di Tomcat. Una procedura classica consiste nel copiare i file direttamente in `webapps`. Tuttavia, la distribuzione di AWS OpsWorks Stacks è progettata per conservare fino a cinque versioni di un'app su un'istanza, quindi puoi tornare a una versione precedente se necessario. AWS OpsWorks Stacks esegue quindi le seguenti operazioni:

1. Distribuisce app in una directory distinta il cui nome include un timestamp, come `/srv/www/my_1st_jsp/releases/20130731141527`.
2. Crea un collegamento simbolico denominato `current`, ad esempio `/srv/www/my_1st_jsp/current`, a questa directory univoca.
3. Se non esiste già, crea un collegamento simbolico dalla directory `webapps` al collegamento simbolico `current` creato al Passaggio 2.

Se devi eseguire il rollback a una versione precedente, modifica il collegamento simbolico `current` in modo che punti a una directory distinta contenente il timestamp appropriato modificando, ad esempio, la destinazione del collegamento di `/srv/www/my_1st_jsp/current`.

La sezione centrale di `tomcat::deploy` configura il collegamento simbolico.

```
 ...
 current_dir = ::File.join(deploy[:deploy_to], 'current')
 webapp_dir = ::File.join(node['tomcat']['webapps_base_dir'],
deploy[:document_root].blank? ? application : deploy[:document_root])

 # opsworks_deploy creates some stub dirs, which are not needed for typical webapps
 ruby_block "remove unnecessary directory entries in #{current_dir}" do
   block do
     node['tomcat']['webapps_dir_entries_to_delete'].each do |dir_entry|
       ::FileUtils.rm_rf(::File.join(current_dir, dir_entry), :secure => true)
     end
   end
 end

 link webapp_dir do
   to current_dir
   action :create
 end
 ...
```

La ricetta prima crea due variabili, `current_dir` e `webapp_dir`, per rappresentare, rispettivamente, le directory `current` e `webapp`. Quindi, utilizza una risorsa `link` per collegare `webapp_dir` a `current_dir`. La `deploy::default` ricetta AWS OpsWorks Stacks crea alcune directory di stub che non sono necessarie per questo esempio, quindi la parte centrale dell'estratto le rimuove.

La parte finale di `tomcat::deploy` riavvia il servizio Tomcat, se necessario.

```
  ...
  include_recipe 'tomcat::service'

  execute 'trigger tomcat service restart' do
    command '/bin/true'
    not_if { node['tomcat']['auto_deploy'].to_s == 'true' }
    notifies :restart, resources(:service => 'tomcat')
  end
end

include_recipe 'tomcat::context'
```

La ricetta innanzitutto esegue `tomcat::service` per garantire che il servizio venga definito per questa esecuzione di Chef. Quindi, utilizza una risorsa execute per notificare il riavvio al servizio, ma solo se `['tomcat']['auto_deploy']` è impostato su `'true'`. In alternativa, Tomcat ascolta le modifiche nella propria directory `webapps`, che rende superfluo un riavvio esplicito del servizio Tomcat.

> **Note**
>
> La risorsa `execute` non esegue effettivamente niente di sostanziale. `/bin/true` è uno script di shell fittizio che restituisce semplicemente un codice di successo. Qui è utilizzato semplicemente come un metodo utile per generare una notifica di riavvio. Come accennato in precedenza, l'utilizzo di notifiche garantisce che i servizi non vengano riavviati troppo frequentemente.

Infine, `tomcat::deploy` esegue `tomcat::context`, che aggiorna il file di configurazione del contesto dell'app Web se hai modificato il database di back-end.

Creare uno stack ed eseguire un'applicazione

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

In questa sezione viene illustrato come utilizzare il libro di ricette Tomcat per implementare una configurazione di base dello stack che esegue una semplice applicazione Java Server Pages (JSP) denominata SimpleJSP. Lo stack è costituito da un livello personalizzato basato su Tomcat denominato e da TomCustom un livello MySQL. SimpleJSP viene distribuito TomCustom e visualizza alcune informazioni dal database MySQL. Se non hai già familiarità con le nozioni di base su come usare AWS OpsWorks Stacks, dovresti prima leggere. Nozioni di base sugli stack Linux Chef 11

Applicazione SimpleJSP

L'applicazione SimpleJSP illustra le basi della configurazione di una connessione di database e del recupero di dati dal database MySQL dello stack.

```
<html>
  <head>
    <title>DB Access</title>
  </head>
  <body>
    <%@ page language="java" import="java.sql.*,javax.naming.*,javax.sql.*" %>
    <%
      StringBuffer output = new StringBuffer();
      DataSource ds = null;
      Connection con = null;
      Statement stmt = null;
      ResultSet rs = null;
      try {
        Context initCtx = new InitialContext();
        ds = (DataSource) initCtx.lookup("java:comp/env/jdbc/mydb");
        con = ds.getConnection();
        output.append("Databases found:<br>");
```

```
        stmt = con.createStatement();
        rs = stmt.executeQuery("show databases");
        while (rs.next()) {
          output.append(rs.getString(1));
          output.append("<br>");
        }
      }
      catch (Exception e) {
        output.append("Exception: ");
        output.append(e.getMessage());
        output.append("<br>");
      }
      finally {
        try {
          if (rs != null) {
            rs.close();
          }
          if (stmt != null) {
            stmt.close();
          }
          if (con != null) {
            con.close();
          }
        }
        catch (Exception e) {
          output.append("Exception (during close of connection): ");
          output.append(e.getMessage());
          output.append("<br>");
        }
      }
    %>
    <%= output.toString() %>
  </body>
</html>
```

SimpleJSP utilizza un oggetto `DataSource` per comunicare con il database MySQL. Tomcat usa i dati nel file di configurazione del contesto dell'app Web per creare e inizializzare un oggetto `DataSource` e associarlo a un nome logico. Quindi, registra il nome logico con un servizio di denominazione JNDI (Java Naming and Directory Interface). Per ottenere un'istanza dell'oggetto `DataSource` appropriato, devi creare un oggetto `InitialContext` e passare il nome logico

della risorsa al metodo `lookup` dell'oggetto, che recupera l'oggetto appropriato. Il nome logico dell'esempio SimpleJSP, `java:comp/env/jdbc/mydb`, ha i seguenti componenti:

- Lo spazio dei nomi radice, `java`, che è separato dal resto del nome da due punti (:).
- Eventuali spazi dei nomi aggiuntivi, separati da barre (/).

  Tomcat aggiunge automaticamente le risorse allo spazio dei nomi `comp/env`.
- Il nome della risorsa, definito nel file di configurazione del contesto dell'app Web e separato dagli spazi dei nomi per mezzo di una barra.

  In questo esempio il nome della risorsa è `jdbc/mydb`.

Per stabilire una connessione al database, SimpleJSP esegue le seguenti operazioni:

1. Chiama il metodo `DataSource` dell'oggetto `getConnection`, che restituisce un oggetto `Connection`.
2. Chiama il metodo `Connection` dell'oggetto `createStatement` per creare un oggetto `Statement`, che puoi utilizzare per comunicare con il database.
3. Comunica con il database chiamando il metodo `Statement` appropriato.

   SimpleJSP chiama `executeQuery` per eseguire una query SHOW DATABASES, che elenca i database del server.

Il metodo `executeQuery` restituisce un oggetto `ResultSet`, che contiene i risultati delle query. SimpleJSP ottiene i nomi dei database dall'oggetto `ResultSet` restituito e li concatena per creare una stringa di output. Infine, l'esempio chiude gli oggetti `ResultSet`, `Statement` e `Connection`. Per ulteriori informazioni su JSP e JDBC, vedere rispettivamente JavaServer Pages Technology e JDBC Basics.

Per usare SimpleJSP con uno stack, devi inserirlo in un repository. Puoi utilizzare uno qualsiasi dei repository supportati, ma per utilizzare SimpleJSP con lo stack di esempio illustrato nella sezione seguente, devi inserirlo in un archivio S3 pubblico. Per informazioni su come utilizzare gli altri repository standard, consulta Repository dei libri di ricette.

Per inserire SimpleJSP in un repository di archiviazione S3

1. Copiare il codice di esempio in un file denominato `simplejsp.jsp` e inserire il file in una directory denominata `simplejsp`.

2. Creare un archivio `.zip` della directory `simplejsp`.
3. Crea un bucket Amazon S3 pubblico, caricalo `simplejsp.zip` nel bucket e rendi pubblico il file.

   Per sapere come eseguire questa attività, consulta Nozioni di base su Amazon Simple Storage Service.

Creare uno stack

Per eseguire SimpleJSP hai bisogno di uno stack con i seguenti livelli.

- Un livello MySQL, che supporta il server MySQL di back-end.
- Un livello personalizzato che utilizza il libro di ricette Tomcat per supportare istanze server Tomcat.

Per creare lo stack

1. Nella dashboard AWS OpsWorks Stacks, fai clic su Aggiungi stack per creare un nuovo stack e fai clic su Avanzato **>>** per visualizzare tutte le opzioni. Configurare lo stack come segue.
   - Nome: un nome di stack definito dall'utente; questo esempio utilizza. TomStack
   - Usa libri di cucina personalizzati per Chef: imposta l'interruttore su Sì, che mostra alcune opzioni aggiuntive.
   - Tipo di repository —Git.
   - URL del repository:. `git://github.com/amazonwebservices/opsworks-example-cookbooks.git`
   - Custom Chef JSON: aggiungi il seguente JSON:

```
{
  "tomcat": {
    "base_version": 7,
    "java_opts": "-Djava.awt.headless=true -Xmx256m"
  },
  "datasources": {
    "ROOT": "jdbc/mydb"
  }
}
```

Per le opzioni rimanenti, puoi accettare le impostazioni predefinite.

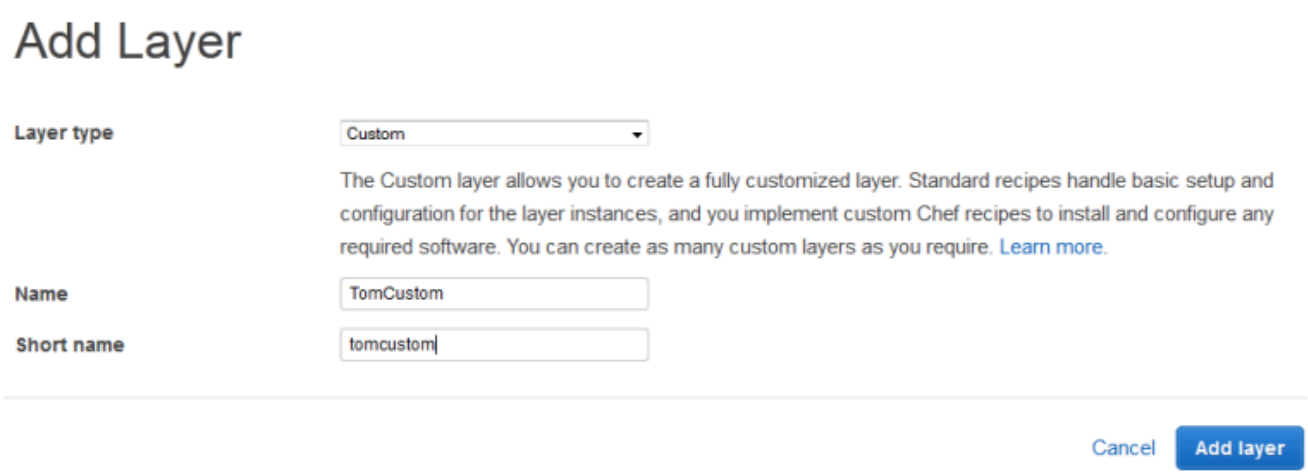
Il JSON personalizzato esegue le operazioni seguenti:

- Ignora l'attributo `['base_version']` del libro di ricette Tomcat per impostare la versione di Tomcat su 7. Il valore predefinito è 6.
- Ignora l'attributo `['java_opts']` del libro di ricette Tomcat per specificare che l'istanza è headless e impostare la dimensione heap massima di JVM su 256 MB. Il valore predefinito non prevede l'impostazione di opzioni per le istanze che eseguono Amazon Linux.
- Specifica il valore dell'attributo `['datasources]`, che assegna un nome di risorsa JDBC (jdbc/mydb) al nome del contesto dell'app Web (ROOT), come descritto in tomcat::context.

  Quest'ultimo attributo non dispone di un valore predefinito, pertanto devi impostarlo con un JSON personalizzato.



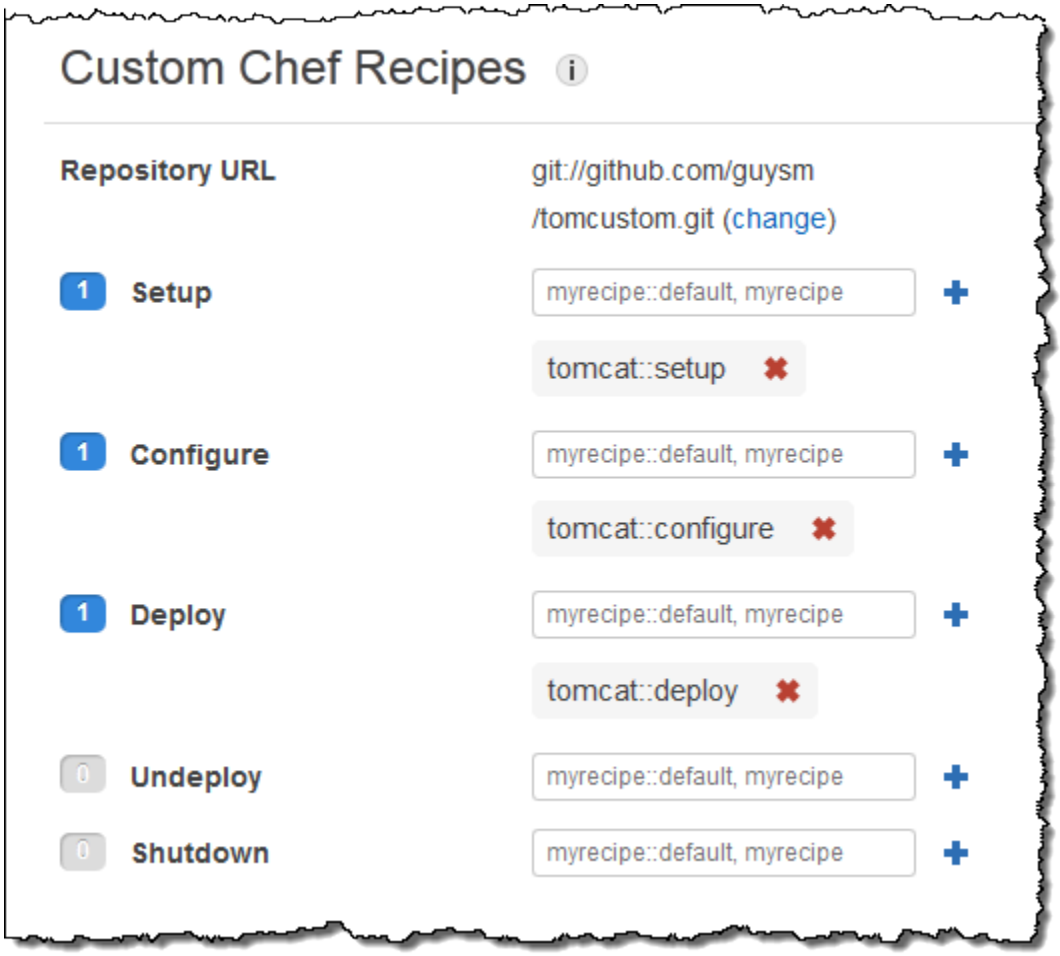
2. Fare clic su Add a layer (Aggiungi un livello). Per Layer type (Tipo di livello), selezionare MySQL. Quindi, fare clic su Add Layer (Aggiungi livello).
3. Fare clic su Instances (Istanze) nel riquadro di navigazione, quindi fare clic su Add an instance (Aggiungi un'istanza). Fare clic su Add Instance (Aggiungi istanza) per accettare le impostazioni predefinite. Sulla riga dell'istanza, fare clic su start (avvia).
4. Tornare alla pagina Layers (Livelli) e fare clic su + Layer (+ Livello) per aggiungere un livello. Per Layer type (Tipo di livello), fare clic su Custom (Personalizzato). L'esempio utilizza **`TomCustom`** e **`tomcustom`**, rispettivamente, come nome e nome abbreviato del livello.



5. Nella pagina Layers (Livelli), per il livello personalizzato, fare clic su Recipes (Ricette), quindi fare clic su Edit (Modifica). In Custom Chef Recipes (Ricette Chef personalizzate), assegnare le ricette del libro di ricette Tomcat agli eventi del ciclo di vita del livello, nel modo seguente:

   - Per Setup (Installazione), digitare **`tomcat::setup`** e fare clic su +.
   - Per Configure (Configurazione), digitare **`tomcat::configure`** e fare clic su +.
   - Per Deploy (Distribuzione), digitare **`tomcat::deploy`** e fare clic su +. Quindi, fare clic su Save (Salva).

   .



6. Fare clic su Apps (App) nel riquadro di navigazione, quindi fare clic su Add an app (Aggiungi un'app). Specificare le seguenti opzioni e fare clic su Add App (Aggiungi app):

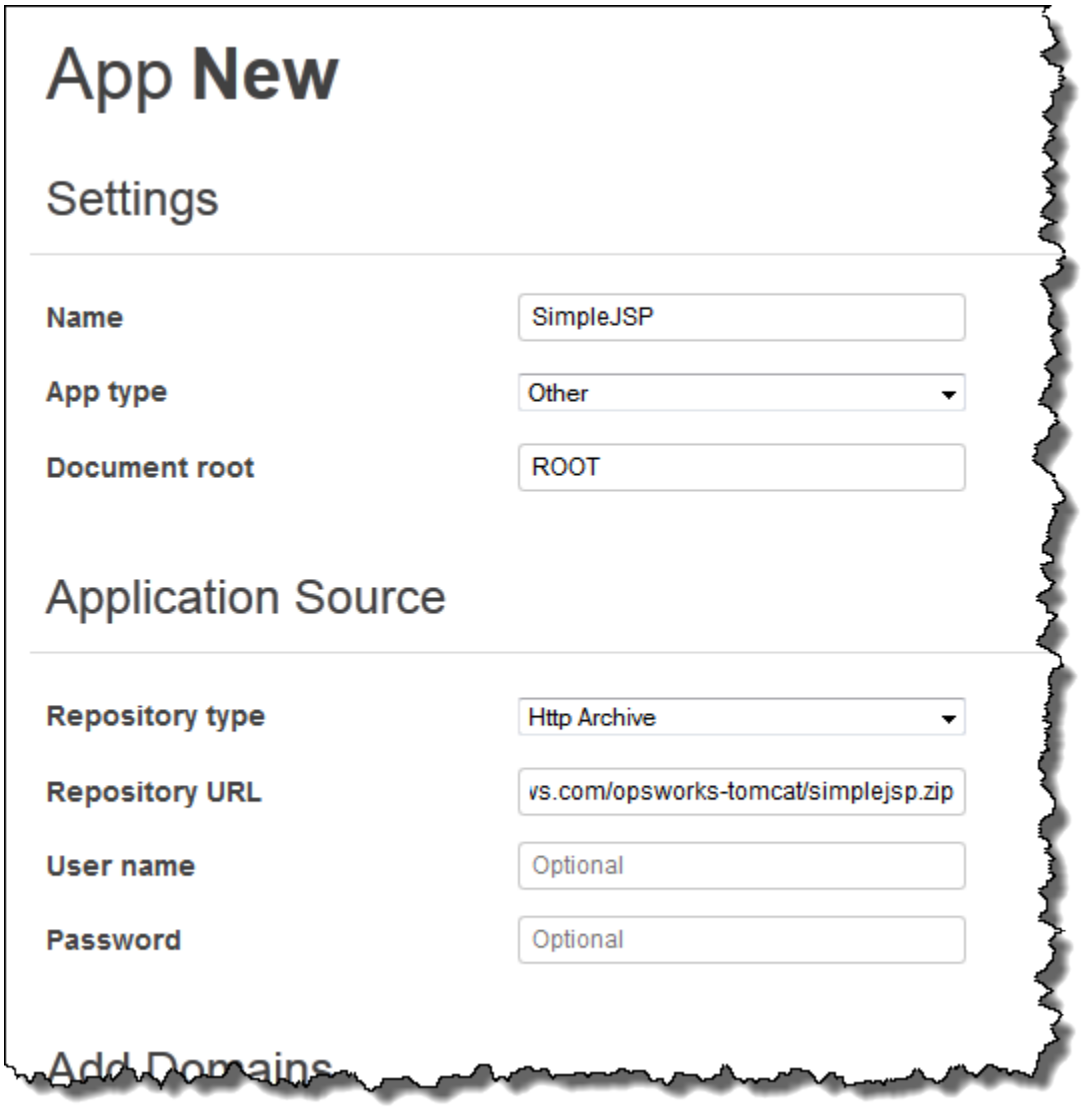
   - Nome: il nome dell'app; l'esempio utilizza SimpleJSP e il nome breve generato da Stacks sarà simplejsp. AWS OpsWorks
   - Tipo di app: imposta questa opzione su Altro.

     AWS OpsWorks Stacks distribuisce automaticamente i tipi di app standard nelle istanze del server associate. Se si imposta App type (Tipo di app) su un'opzione diversa, AWS OpsWorks Stacks esegue semplicemente le ricette di distribuzione e consente loro di gestire la distribuzione.
   - Document root: imposta questa opzione su. **ROOT**

     Il valore Document root (Radice documento) specifica il nome del contesto.
   - Tipo di repository: imposta questa opzione su S3 Archive.

- URL del repository: imposta questo valore sull'URL Amazon S3 dell'app che hai creato in precedenza.

Utilizzare le impostazioni predefinite per le altre opzioni.



7. Utilizza la pagina Istanze per aggiungere un'istanza al TomCustom layer e avviarla. AWS OpsWorks Stacks esegue automaticamente le ricette Deploy su una nuova istanza al termine delle ricette di installazione, quindi l'avvio dell'istanza distribuisce anche SimpleJSP.
8. Quando l' TomCustom istanza è online, fai clic sul nome dell'istanza nella pagina Istanze per visualizzarne i dettagli. Copiare l'indirizzo IP pubblico. Quindi costruisci un URL come segue: http://*publicIP*/tc/. *appname.jsp* Ad esempio, l'URL sarà simile al seguente: **`http://50.218.191.172/tc/simplejsp.jsp`**.

> **Note**
>
> L'URL Apache che inoltra le richieste a Tomcat è impostato sull'attributo `['tomcat']['apache_tomcat_bind_path']` predefinito, `/tc/`. La radice del documento SimpleJSP è impostata su `ROOT`, un valore speciale che si risolve su `/`. L'URL è quindi ".../tc/simplejsp.jsp".

9. Incollare nel browser l'URL copiato al passaggio precedente. Verrà visualizzato un codice analogo al seguente:

```
Databases found:
information_schema
simplejsp
test
```

> **Note**
>
> Se il tuo stack ha un'istanza MySQL AWS OpsWorks , Stacks crea automaticamente un database per ogni app, denominato con il nome breve dell'app.

Attributi di configurazione e distribuzione dello stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Quando AWS OpsWorks Stacks esegue un comando su un'istanza, ad esempio un comando deploy in risposta a un evento del ciclo di vita Deploy, aggiunge un set di attributi all'oggetto nodo dell'istanza che descrive la configurazione corrente dello stack. Per gli eventi Deploy e i comandi dello stack Execute Recipes, Stacks installa gli attributi deploy, che forniscono alcune informazioni aggiuntive

sulla distribuzione. AWS OpsWorks Per ulteriori informazioni sull'oggetto nodo, consulta Sostituzione degli attributi. Per un elenco degli attributi di configurazione e distribuzione dello stack più usati, inclusi i nomi di nodo completi, consulta Attributi di configurazione e distribuzione dello stack: Linux e Attributi dei libri di ricette predefiniti.

> **Note**
>
> Negli stack Linux, è possibile ottenere un elenco completo di questi attributi, formattati come oggetto JSON, utilizzando il comando get_json dell'interfaccia a riga di comando dell'agente.

Le seguenti sezioni mostrano gli attributi associati a un evento Configure (Configurazione) e a un evento Deploy (Distribuzione) per uno stack semplice, costituito da quanto segue:

- Un livello PHP App Server con due istanze
- Un HAProxy livello con un'istanza

Gli esempi provengono da una delle istanze di PHP App Server, php-app1. Per comodità, gli attributi sono formattati come oggetto JSON. La struttura dell'oggetto è mappata ai nomi completi degli attributi. Ad esempio, l'attributo `node[:opsworks][:ruby_version]` viene visualizzato come segue in una rappresentazione JSON.

```
{
  "opsworks": {
    ...
    "ruby_version": "1.8.7",
    ...
  }
}
```

Argomenti

- Attributi di configurazione
- Attributi di distribuzione

Attributi di configurazione

L'oggetto JSON seguente mostra gli attributi per un evento Configure (Configurazione), che si verifica in ogni istanza in uno stack quando un'istanza passa online o offline. Gli attributi includono gli attributi di configurazione dello stack predefiniti ed eventuali attributi JSON personalizzati definiti per lo stack prima dell'evento (nessuno in questo esempio). L'esempio è stato modificato per contenere la lunghezza. Per una descrizione dettagliata dei diversi attributi, consulta Attributi di configurazione e distribuzione dello stack: Linux e Attributi dei libri di ricette predefiniti.

```
{
  "opsworks": {
    "layers": {
      "php-app": {
        "id": "4a2a56c8-f909-4b39-81f8-556536d20648",
        "instances": {
          "php-app2": {
            "elastic_ip": null,
            "region": "us-west-2",
            "booted_at": "2013-02-26T20:41:10+00:00",
            "ip": "192.0.2.0",
            "aws_instance_id": "i-34037f06",
            "availability_zone": "us-west-2a",
            "instance_type": "c1.medium",
            "private_dns_name": "ip-10-252-0-203.us-west-2.compute.internal",
            "private_ip": "10.252.0.203",
            "created_at": "2013-02-26T20:39:39+00:00",
            "status": "online",
            "backends": 8,
            "public_dns_name": "ec2-192-0-2-0.us-west-2.compute.amazonaws.com"
          },
          "php-app1": {
            ...
          }
        },
        "name": "PHP Application Server"
      },
      "lb": {
        "id": "15c86142-d836-4191-860f-f4d310440f14",
        "instances": {
          "lb1": {
           ...
          }
```

```
        },
        "name": "Load Balancer"
      }
    },
    "agent_version": "104",
    "applications": [

    ],
    "stack": {
      "name": "MyStack"
    },
    "ruby_version": "1.8.7",
    "sent_at": 1361911623,
    "ruby_stack": "ruby_enterprise",
    "instance": {
      "layers": [
        "php-app"
      ],
      "region": "us-west-2",
      "ip": "192.0.2.0",
      "id": "45ef378d-b87c-42be-a1b9-b67c48edafd4",
      "aws_instance_id": "i-32037f00",
      "availability_zone": "us-west-2a",
      "private_dns_name": "ip-10-252-84-253.us-west-2.compute.internal",
      "instance_type": "c1.medium",
      "hostname": "php-app1",
      "private_ip": "10.252.84.253",
      "backends": 8,
      "architecture": "i386",
      "public_dns_name": "ec2-192-0-2-0.us-west-2.compute.amazonaws.com"
    },
    "activity": "configure",
    "rails_stack": {
      "name": null
    },
    "deployment": null,
    "valid_client_activities": [
      "reboot",
      "stop",
      "setup",
      "configure",
      "update_dependencies",
      "install_dependencies",
      "update_custom_cookbooks",
```

```
      "execute_recipes"
    ]
  },
  "opsworks_custom_cookbooks": {
    "recipes": [

    ],
    "enabled": false
  },
  "recipes": [
    "opsworks_custom_cookbooks::load",
    "opsworks_ganglia::configure-client",
    "ssh_users",
    "agent_version",
    "mod_php5_apache2::php",
    "php::configure",
    "opsworks_stack_state_sync",
    "opsworks_custom_cookbooks::execute",
    "test_suite",
    "opsworks_cleanup"
  ],
  "opsworks_rubygems": {
    "version": "1.8.24"
  },
  "ssh_users": {
  },
  "opsworks_bundler": {
    "manage_package": null,
    "version": "1.0.10"
  },
  "deploy": {
  }
}
```

La maggior parte delle informazioni è inclusa nell'attributo `opsworks`, che viene spesso definito come spazio dei nomi. L'elenco seguente descrive gli attributi principali:

- `layers`attributi: un insieme di attributi, ognuno dei quali descrive la configurazione di uno dei livelli dello stack.

  Per questo esempio i livelli sono identificati dai relativi nomi brevi `php-app` e `lb`. Per ulteriori informazioni sui nomi brevi per altri livelli, consulta AWS OpsWorks Stacks Layer Reference.

- `instances`attributi: ogni livello ha un `instances` elemento, che include un attributo per ciascuna istanza online dei livelli, denominato con il nome breve dell'istanza.

  Il livello PHP App Server ha due istanze e. `php-app1` `php-app2` Il HAProxy livello ha un'istanza,. `lb1`

  > **Note**
  >
  > L'elemento `instances` contiene solo le istanze che si trovano nello stato online quando vengono creati gli attributi di configurazione e dello stack specifici.

- Attributi dell'istanza: ogni attributo dell'istanza contiene un set di attributi che caratterizzano l'istanza, come l'indirizzo IP privato e il nome DNS privato dell'istanza. Per brevità, l'esempio mostra solo l'attributo `php-app2` in dettaglio. Gli altri contengono informazioni simili.
- `applications`— Un elenco di app distribuite, non utilizzate in questo esempio.
- `stack`— Il nome dello stack; `MyStack` in questo esempio.
- `instance`— L'istanza in cui sono installati questi attributi; `php-app1` in questo esempio. Le ricette possono utilizzare questo attributo per ottenere informazioni sull'istanza in cui sono in esecuzione, ad esempio l'indirizzo IP pubblico dell'istanza.
- `activity`— L'attività che ha prodotto gli attributi; un evento Configure in questo esempio.
- `rails_stack`— Lo stack Rails per gli stack che includono un livello Rails App Server.
- `deployment`— Se questi attributi sono associati a una distribuzione. In questo esempio è impostato su `null` perché gli attributi sono associati a un evento Configure (Configurazione).
- `valid_client_activities`— Un elenco di attività client valide.

L'attributo `opsworks` è seguito da alcuni altri attributi di primo livello, tra cui i seguenti:

- `opsworks_custom_cookbooks`— Se i libri di cucina personalizzati sono abilitati. In caso affermativo, l'attributo include un elenco di ricette personalizzate.
- `recipes`— Le ricette gestite da questa attività.
- `opsworks_rubygems`— La RubyGems versione dell'istanza.
- `ssh_users`— Un elenco di utenti SSH; nessuno in questo esempio.
- `opsworks_bundler`— La versione del bundler e se è abilitata.
- `deploy`— Informazioni sulle attività di distribuzione; nessuna in questo esempio.

Attributi di distribuzione

Gli attributi per un evento Deploy (Distribuzione) o un comando dello stack Execute Recipes (Esegui ricette) sono costituiti dagli attributi di configurazione e distribuzione dello stack predefiniti e da eventuali attributi dello stack o di distribuzione personalizzati (nessuno per questo esempio). L'oggetto JSON seguente mostra gli attributi di php-app1 associati a un evento Deploy (Distribuzione) che ha distribuito l'app SimplePHP nelle istanze PHP dello stack. La maggior parte dell'oggetto è costituita da attributi di configurazione dello stack simili a quelli per l'evento Configure (Configurazione) descritto nella sezione precedente, quindi l'esempio è incentrato principalmente sugli attributi specifici della distribuzione. Per una descrizione dettagliata dei diversi attributi, consulta Attributi di configurazione e distribuzione dello stack: Linux e Attributi dei libri di ricette predefiniti.

```
{
   ...
  "opsworks": {
    ...
    "activity": "deploy",
    "applications": [
      {
        "slug_name": "simplephp",
        "name": "SimplePHP",
        "application_type": "php"
      }
    ],
    "deployment": "5e6242d7-8111-40ee-bddb-00de064ab18f",
    ...
  },
  ...
{
  "ssh_users": {
  },
  "deploy": {
    "simplephpapp": {
      "application": "simplephpapp",
      "application_type": "php",
      "environment_variables": {
        "USER_ID": "168424",
        "USER_KEY": "somepassword"
      },
      "auto_bundle_on_deploy": true,
      "deploy_to": "/srv/www/simplephpapp",
```

```
      "deploying_user": "arn:aws:iam::123456789012:user/guysm",
      "document_root": null,
      "domains": [
        "simplephpapp"
      ],
      "migrate": false,
      "mounted_at": null,
      "rails_env": null,
      "restart_command": "echo 'restarting app'",
      "sleep_before_restart": 0,
      "ssl_support": false,
      "ssl_certificate": null,
      "ssl_certificate_key": null,
      "ssl_certificate_ca": null,
      "scm": {
        "scm_type": "git",
        "repository": "git://github.com/amazonwebservices/opsworks-demo-php-simple-
app.git",
        "revision": "version1",
        "ssh_key": null,
        "user": null,
        "password": null
      },
      "symlink_before_migrate": {
        "config/opsworks.php": "opsworks.php"
      },
      "symlinks": {
      },
      "database": {
      },
      "memcached": {
        "host": null,
        "port": 11211
      },
      "stack": {
        "needs_reload": false
      }
    }
  },
}
```

L'attributo `opsworks` è in gran parte identico all'esempio nella sezione precedente. Le sezioni seguenti sono più rilevanti per la distribuzione:

- `activity`— L'evento associato a questi attributi; un evento Deploy in questo esempio.
- `applications`— Contiene un set di attributi per ogni app che forniscono i nomi, i nomi degli slug e i tipi delle app.

  Il nome dello slug è il nome breve dell'app, che AWS OpsWorks Stacks genera dal nome dell'app. Il nome comune per SimplePHP è simplephp.
- `deployment`— L'ID di distribuzione, che identifica in modo univoco una distribuzione.

L'attributo `deploy` include informazioni sulle app che vengono distribuite. Ad esempio, le ricette di Deploy (Distribuzione) predefinite utilizzano i dati nell'attributo `deploy` per installare i file nelle directory appropriate e creare i file di connessione al database. L'attributo `deploy` include un attributo per ogni app distribuita, il cui nome dipende dal nome breve dell'app. Ogni attributo dell'app include gli attributi seguenti:

- `environment_variables`— Contiene tutte le variabili di ambiente definite per l'app. Per ulteriori informazioni, consulta Variabili di ambiente.
- `domains`— Per impostazione predefinita, il dominio è il nome breve dell'app, che in questo esempio è simplephpapp. Se hai assegnato domini personalizzati, vengono visualizzati anch'essi in questa posizione. Per ulteriori informazioni, consulta Utilizzo di domini personalizzati.
- `application`— Il nome breve dell'app.
- `scm`— Questo elemento contiene le informazioni necessarie per scaricare i file dell'app dal suo repository; un repository Git in questo esempio.
- `database`— Informazioni sul database, se lo stack include un livello di database.
- `document_root`— La radice del documento, impostata su `null` in questo esempio, indica che la radice è pubblica.
- `ssl_certificate_ca`,`ssl_support`, `ssl_certificate_key` — Indica se l'app supporta SSL. In caso affermativo, gli attributi `ssl_certificate_key` e `ssl_certificate_ca` sono impostati sui certificati corrispondenti.
- `deploy_to`— La directory principale dell'app.

Introduzione ai libri di ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Uno stack AWS OpsWorks Stacks a livello di produzione richiede in genere alcune personalizzazioni, il che spesso significa implementare un ricettario Chef personalizzato con una o più ricette, file di attributi o file modello. Questo argomento è un tutorial introduttivo all'implementazione di libri di cucina per Stacks. AWS OpsWorks

Per ulteriori informazioni su come AWS OpsWorks Stacks utilizza i libri di cucina, che include una breve introduzione generale ai libri di cucina, vedi. Libri di ricette e ricette Per ulteriori informazioni su come implementare e testare ricette Chef, consulta la pagina relativa all'infrastruttura basata sui test con Chef, seconda edizione.

Gli esempi del tutorial sono suddivisi in due sezioni:

- Nozioni di base sui libri di ricette è un set di procedure guidate di esempio destinate agli utenti che non hanno familiarità con Chef. Gli utenti esperti di Chef possono ignorare questa sezione.

  Gli esempi presentano le nozioni di base su come implementare libri di ricette per eseguire attività comuni, tra cui l'installazione di pacchetti o la creazione di directory. Per semplificare il processo, potrai utilizzare un paio di utili strumenti, Vagrant e Test Kitchen, per eseguire la maggior parte degli esempi in locale in una macchina virtuale. Prima di iniziare Nozioni di base sui libri di ricette, consulta Vagrant e Test Kitchen per scoprire come installare e utilizzare questi strumenti. Poiché Test Kitchen non supporta ancora Windows, gli esempi sono tutti destinati a Linux, con note che indicano come adattarli a Windows.
- Implementazione di Cookbooks for Stacks AWS OpsWorksdescrive come implementare le ricette per AWS OpsWorks Stacks, anche per gli stack di Windows.

  Include anche alcuni suggerimenti più avanzati, come usare Berkshelf per gestire libri di cucina esterni. Gli esempi sono stati scritti per i nuovi utenti di Chef, proprio come gli esempi contenuti in

Nozioni di base sui libri di ricette. Tuttavia AWS OpsWorks Stacks funziona in modo leggermente diverso rispetto al server Chef, quindi consigliamo agli utenti esperti di Chef di leggere almeno questa sezione.

Vagrant e Test Kitchen

Se utilizzi ricette per istanze Linux, Vagrant e Test Kitchen sono strumenti molto utili per la formazione e per lo sviluppo e i test iniziali. Questo fornisce brevi descrizioni di Vagrant e Test Kitchen e ti indirizza alle istruzioni di installazione e alle procedure dettagliate che ti consentiranno di configurare e familiarizzare con le basi su come utilizzare gli strumenti. Poiché Vagrant supporta Windows, ma Test Kitchen no, vengono forniti solo esempi Linux per questi strumenti.

Vagrant

Vagrant offre un ambiente uniforme per eseguire e testare codice in una macchina virtuale. Supporta un'ampia varietà di ambienti, chiamati Vagrant box, ognuno dei quali rappresenta un sistema operativo configurato. Per AWS OpsWorks Stacks, gli ambienti di interesse si basano su distribuzioni Ubuntu, Amazon o Red Hat Enterprise Linux (RHEL), quindi gli esempi utilizzano principalmente una casella Vagrant denominata. `opscode-ubuntu-12.04`

Poiché Vagrant è disponibile per sistemi Linux, Windows e Macintosh, puoi utilizzare la workstation che preferisci per implementare e testare ricette in qualsiasi sistema operativo supportato. Gli esempi di questo capitolo sono stati creati su un sistema Ubuntu Linux, ma la traduzione delle procedure in sistemi Windows o Macintosh è semplice.

Vagrant è fondamentalmente un wrapper per un provider di virtualizzazione. La maggior parte degli esempi utilizza il provider. VirtualBox VirtualBox è gratuito e disponibile per i sistemi Linux, Windows e Macintosh. La procedura dettagliata di Vagrant fornisce istruzioni di installazione se non le hai già sul tuo sistema. VirtualBox Tieni presente che puoi eseguire ambienti basati su Ubuntu, VirtualBox ma Amazon Linux è disponibile solo per le istanze Amazon. EC2 Tuttavia, è possibile eseguire un sistema operativo simile come CentOS on VirtualBox, utile per lo sviluppo e il test iniziali.

Per informazioni su altri provider, consulta la documentazione di Vagrant. In particolare, il fornitore di `vagrant-aws` plug-in consente di utilizzare Vagrant con istanze Amazon EC2 . Questo provider è particolarmente utile per testare ricette su Amazon Linux, che è disponibile solo su EC2 istanze Amazon. Il provider `vagrant-aws` è gratuito, ma devi avere un account AWS e pagare per tutte le risorse AWS che utilizzi.

A questo punto, segui la procedura dettagliata introduttiva di Vagrant, che descrive come installare Vagrant nella workstation e fornisce le nozioni di base su come utilizzare Vagrant. Tieni presente che gli esempi forniti in questo capitolo non utilizzano un repository Git e quindi puoi omettere questa parte della procedura guidata se lo desideri.

Test Kitchen

Test Kitchen semplifica il processo di esecuzione e test dei libri di ricette in Vagrant. In pratica, quasi mai dovrai utilizzare Vagrant direttamente. Test Kitchen esegue la maggior parte delle attività comuni, tra cui:

- Avvio di un'istanza in Vagrant.
- Trasferimento di libri di ricette all'istanza.
- Esecuzione delle ricette del libro di ricette nell'istanza.
- Test delle ricette di un libro di ricette nell'istanza.
- Utilizzo di SSH per accedere all'istanza.

Invece di installare il pacchetto gem Test Kitchen direttamente, ti consigliamo di installare Chef DK. Oltre allo stesso Chef, questo pacchetto include Test Kitchen, Berkshelf e molti altri strumenti utili. ChefSpec

A questo punto, ti consigliamo di seguire la procedura guidata introduttiva di Test Kitchen, che fornisce le nozioni di base su come utilizzare Test Kitchen per eseguire e testare ricette.

> **Note**
>
> Gli esempi inclusi in questo capitolo utilizzano Test Kitchen come pratico metodo di esecuzione di ricette. Se preferisci, puoi interrompere la procedura dettagliata introduttiva dopo aver completato la sezione relativa alla verifica manuale, che descrive tutto quello che devi sapere per gli esempi. Tuttavia, Test Kitchen è prevalentemente una piattaforma di test che supporta framework di test come Bash Automated Test System (BATS). A un certo punto, dovrai completare la parte restante della procedura guidata per scoprire come utilizzare Test Kitchen per testare le ricette.

Nozioni di base sui libri di ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

I libri di ricette sono perfetti per eseguire un'ampia gamma di attività. Gli argomenti seguenti presuppongono che tu non abbia mai utilizzato Chef e descrivono come usare i libri di ricette per eseguire alcune operazioni comuni. Poiché Test Kitchen non supporta ancora Windows, gli esempi sono tutti destinati a Linux, con note che indicano come adattarli a Windows. Se non hai mai utilizzato Chef, ti consigliamo di seguire questi esempi anche se lavori con Windows. La maggior parte degli esempi inclusi in questo argomento può essere utilizzata su istanze Windows con alcune lievi modifiche (annotate negli esempi). Tutti gli esempi vengono eseguiti su una macchina virtuale, perciò non è necessario disporre di un computer Linux. Ti basta installare Vagrant e Test Kitchen sulla tua normale workstation.

> **Note**
>
> Se vuoi eseguire queste ricette su un'istanza di Windows, l'approccio più semplice è creare uno stack Windows ed eseguire le ricette su una delle istanze dello stack. Per ulteriori informazioni su come eseguire ricette su un'istanza AWS OpsWorks Stacks Windows, consulta. Esecuzione di una ricetta su un'istanza Windows

Prima di continuare, assicurati di aver installato Vagrant e Test Kitchen e di aver completato le rispettive procedure guidate sulle nozioni di base. Per ulteriori informazioni, consulta Vagrant e Test Kitchen.

Argomenti

- Struttura di una ricetta
- Esempio 1: installazione di pacchetti
- Esempio 2: gestione degli utenti

- Esempio 3: creazione di directory
- Esempio 4: aggiunta del controllo del flusso
- Esempio 5: utilizzo di attributi
- Esempio 6: creazione di file
- Esempio 7: esecuzione di comandi e script
- Esempio 8: gestione dei servizi
- Esempio 9: utilizzo di Amazon EC2 Instances
- Fasi successive

Struttura di una ricetta

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Un libro di ricette è principalmente un set di ricette, in grado di eseguire un'ampia gamma di attività su un'istanza. Per chiarire come funziona l'implementazione delle ricette, è utile guardare un semplice esempio. Di seguito è riportata la ricetta per il livello HAProxy integrato. In questa fase soffermati solo sulla struttura generale, senza preoccuparti troppo dei dettagli, che saranno descritti nei successivi esempi.

```
package 'haproxy' do
  action :install
end

if platform?('debian','ubuntu')
  template '/etc/default/haproxy' do
    source 'haproxy-default.erb'
    owner 'root'
    group 'root'
    mode 0644
```

```
  end
end

include_recipe 'haproxy::service'

service 'haproxy' do
  action [:enable, :start]
end

template '/etc/haproxy/haproxy.cfg' do
  source 'haproxy.cfg.erb'
  owner 'root'
  group 'root'
  mode 0644
  notifies :restart, "service[haproxy]"
end
```

> **Note**
>
> Per questo e altri esempi di utilizzo di ricette di lavoro e file correlati, consulta le ricette integrate di AWS OpsWorks Stacks.

L'esempio evidenzia gli elementi chiave della ricetta, che sono descritti nelle seguenti sezioni.

Argomenti

- Risorse
- Controllo di flusso
- Ricette incluse

Risorse

Le Ricette consistono in gran parte in un set di risorse Chef. Ognuna di esse specifica un particolare aspetto dello stato finale dell'istanza, ad esempio un pacchetto da installare o un servizio da avviare. L'esempio presenta quattro risorse:

- Una `package` risorsa, che rappresenta un pacchetto installato, un HAProxy server per questo esempio.
- Una `service` risorsa, che rappresenta un servizio, il HAProxy servizio di questo esempio.

- Due `template` risorse, che rappresentano file che devono essere creati da un modello specificato, due file di HAProxy configurazione per questo esempio.

Le risorse forniscono un modo dichiarativo per specificare l'istanza. In background, a ogni risorsa è associato un provider che esegue le attività necessarie, ad esempio l'installazione di pacchetti, la creazione e la configurazione di directory, l'avvio di servizi e così via. Se i dettagli dell'attività dipendono dallo specifico sistema operativo, la risorsa ha più provider e usa il tipo appropriato per il sistema. Ad esempio, su un sistema Red Hat Linux, il provider `package` utilizza `yum` per l'installazione di pacchetti. In un sistema Ubuntu Linux, il provider `package` utilizza `apt-get`.

Una risorsa come un blocco di codice Ruby viene implementata con il seguente formato generale.

```
resource_type "resource_name" do
  attribute1 'value1'
  attribute2 'value2'
  ...
  action :action_name
  notifies : action 'resource'
end
```

Gli elementi sono:

Tipo di risorsa

(Obbligatorio) L'esempio include tre tipi di risorse: `package`, `service` e `template`.

Nome risorsa

(Obbligatorio) Il nome identifica la particolare risorsa e talvolta viene usato come valore di default per uno degli attributi. Nell'esempio, `package` rappresenta una risorsa di pacchetto denominata `haproxy` e la prima risorsa `template` rappresenta un file di configurazione denominato `/etc/default/haproxy`.

Attributes

(Facoltativo) Gli attributi specificano la configurazione delle risorse e variano in base al tipo di risorsa e alla sua configurazione.

- Le risorse `template` dell'esempio definiscono esplicitamente una serie di attributi che specifica l'origine del file creato nonché il proprietario, il gruppo e la modalità.

- Le risorse `package` e `service` esempio non definiscono in modo esplicito alcun attributo.

  Il nome della risorsa in genere è il valore di default per un attributo obbligatorio ed è talvolta tutto ciò che è necessario. Ad esempio, il nome della risorsa è il valore di default per l'attributo `package` della risorsa `package_name`, che è l'unico attributo obbligatorio.

Ci sono anche alcuni attributi specializzati, denominati attributi di protezione, che specificano quando il provider di risorse deve eseguire un'azione. Ad esempio, l'attributo `only_if` indica al provider di risorse di eseguire un'azione solo se viene soddisfatta una condizione specificata. La HAProxy ricetta non utilizza gli attributi guard, ma vengono utilizzati in molti dei seguenti esempi.

Operazioni e notifiche

(Facoltativo) Azioni e notifiche specificano quali attività deve eseguire il provider.

- `action` indica al provider di eseguire un'operazione specifica, ad esempio un'installazione o una creazione.

  Ogni risorsa dispone di un set di operazioni che dipendono da quella particolare risorsa, una delle quali è l'operazione predefinita. Nell'esempio, l'operazione della risorsa `package` è `install`, che indica al provider di installare il pacchetto. La prima risorsa `template` non include alcun elemento `action`, perciò il provider esegue l'operazione predefinita `create`.
- `notifies` indica un altro provider della risorsa di eseguire un'operazione, ma solo se lo stato della risorsa è stato modificato.

  `notifies` viene in genere utilizzato con risorse quali `template` e `file` per eseguire attività come il riavvio di un servizio dopo la modifica di un file di configurazione. Le risorse non prevedono notifiche di default. Per le notifica, la risorsa deve avere un elemento `notifies` esplicito. Nella HAProxy ricetta, la seconda `template` risorsa notifica alla risorsa haproxy `service` di riavviare il HAProxy servizio se il file di configurazione associato è cambiato.

Le risorse talvolta dipendono dal sistema operativo.

- Alcune risorse possono essere utilizzate solo su sistemi Linux o Windows.

  Ad esempio, package installa pacchetti sui sistemi Linux e windows_package installa pacchetti sui sistemi Windows.
- Alcune risorse possono essere utilizzate con qualsiasi sistema operativo, ma dispongono di attributi specifici per un determinato sistema.

Ad esempio, la risorsa file può essere utilizzata sia su sistemi Linux che Windows, ma dispone di set di attributi separati per la configurazione delle autorizzazioni.

Per le descrizioni delle risorse standard, inclusi gli attributi, le operazioni e le notifiche disponibili per ciascuna risorsa, consulta la pagina relativa alle informazioni su risorse e provider.

Controllo di flusso

Poiché le ricette sono applicazioni Ruby, è possibile utilizzare le strutture di controllo di Ruby per integrare il controllo del flusso in una ricetta. Ad esempio, è possibile usare logica condizionale di Ruby per far sì che la ricetta si comporti in modo diverso su sistemi differenti. La HAProxy ricetta include un `if` blocco che utilizza una `template` risorsa per creare un file di configurazione, ma solo se la ricetta è in esecuzione su un sistema Debian o Ubuntu.

È comune anche lo scenario che prevede l'uso di un loop per eseguire una risorsa più volte con diverse impostazioni di attributo. Ad esempio, è possibile creare un set di directory utilizzando un loop per eseguire una risorsa `directory` più volte con diversi nomi di directory.

> **Note**
>
> Se non hai familiarità con Ruby, consulta l'articolo sulle nozioni di base su Ruby per Chef, che illustra le informazioni necessarie per la maggior parte delle ricette.

Ricette incluse

`include_recipe` include altre ricette nel codice per permetterti di suddividere in moduli le ricette e riutilizzare lo stesso codice in più ricette. Quando esegui la ricetta host, Chef sostituisce ciascun elemento `include_recipe` con il codice della ricetta specificato prima di eseguire la ricetta host. Puoi identificare una ricetta inclusa utilizzando la sintassi standard `cookbook_name::recipe_name` di Chef, dove `recipe_name` omette l'estensione `.rb`. L'esempio include una ricetta`haproxy::service`, che rappresenta il HAProxy servizio.

> **Note**
>
> Se utilizzi `include_recipe` in ricette in esecuzione su Chef 11.10 e versioni successive per includere una ricetta da un altro libro di ricette, devi utilizzare un'istruzione `depends` per

dichiarare la dipendenza nel file `metadata.rb` del libro di ricette. Per ulteriori informazioni, consulta Implementazione delle ricette: Chef 11.10.

Esempio 1: installazione di pacchetti

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

L'installazione di un pacchetto è uno dei casi d'uso più comuni per le ricette e può essere molto semplice, a seconda del pacchetto. Ad esempio, la seguente ricetta installa Git su un sistema Linux.

```
package 'git' do
  action :install
end
```

La risorsa `package` gestisce l'installazione di pacchetti. Per questo esempio, non è necessario specificare attributi. Il nome della risorsa è il valore di default per l'attributo `package_name` che identifica il pacchetto. L'operazione `install` indica al provider di installare il pacchetto. È possibile rendere il codice ancora più semplice saltando `install`; è l'operazione predefinita della risorsa `package`. Quando esegui la ricetta, Chef usa il provider appropriato per installare il pacchetto. Nel sistema Ubuntu che utilizzerai per l'esempio, il provider installa Git chiamando `apt-get`.

> **Note**
>
> L'installazione di software su un sistema Windows richiede una procedura diversa. Per ulteriori informazioni, consulta Installazione di software Windows.

Per utilizzare Test Kitchen per eseguire questa ricetta in Vagrant, devi prima impostare un libro di ricette e inizializzare e configurare Test Kitchen. Quanto segue è relativo a un sistema Linux, ma la

procedura è essenzialmente simile per i sistemi Windows e Macintosh. Inizia aprendo una finestra del terminale, perché tutti gli esempi in questo capitolo utilizzano strumenti a riga di comando.

Per preparare il libro di ricette

1. Nella directory principale, creare una sottodirectory denominata `opsworks_cookbooks`, che conterrà tutti i libri di ricette per questo capitolo. Creare quindi una sottodirectory per questo libro di ricette denominata `installpkg` e accedervi.
2. In `installpkg`, creare un file denominato `metadata.rb` che contiene il codice seguente.

   ```
   name "installpkg"
   version "0.1.0"
   ```

   Per semplicità, gli esempi in questo capitolo specificano solo il nome e la versione del libro di ricette, ma `metadata.rb` può contenere un'ampia gamma di metadati del libro di ricette. Per ulteriori informazioni, consulta l'articolo sulle informazioni sui metadati dei libri di ricette.

   > **Note**
   >
   > Assicurati di creare `metadata.rb` prima di inizializzare Test Kitchen, perché utilizza i dati per creare il file di configurazione di default.

3. In `installpkg`, eseguire `kitchen init`, che inizializza Test Kitchen e installa il driver Vagrant di default.
4. Il comando `kitchen init` crea un file di configurazione YAML in `installpkg` denominato `.kitchen.yml`. Aprire il file nell'editor di testo preferito. Il file `.kitchen.yml` include una sezione `platforms` che specifica su quali sistemi eseguire le ricette. Test Kitchen crea un'istanza ed esegue le ricette specificate su ciascuna piattaforma.

   > **Note**
   >
   > Per impostazione predefinita, Test Kitchen esegue le ricette una piattaforma alla volta. Aggiungendo un argomento `-p` a qualsiasi comando che crea un'istanza, Test Kitchen eseguirà le ricette su ogni piattaforma, in parallelo.

   Una singola piattaforma è sufficiente per questo esempio, quindi modificare `.kitchen.yml` per rimuovere la piattaforma `centos-6.4`. Il file `.kitchen.yml` dovrebbe ora apparire così:

```
---
driver:
  name: vagrant

provisioner:
  name: chef_solo

platforms:
  - name: ubuntu-12.04

suites:
  - name: default
    run_list:
      - recipe[installpkg::default]
    attributes:
```

Test Kitchen esegue solo le ricette incluse nell'elenco di esecuzione `.kitchen.yml`. Puoi identificare le ricette utilizzando il `[`*`cookbook_name`*`::`*`recipe_name`*`]` formato, dove *`recipe_name`* omette l'estensione. `.rb` Inizialmente, l'elenco di esecuzione `.kitchen.yml` contiene la ricetta predefinita del libro di ricette, `installpkg::default`. Questa è la ricetta da implementare, perciò non è necessario modificare l'elenco di esecuzione.

5. Crea una sottodirectory di `installpkg` denominata `recipes`.

   Se un libro di cucina contiene ricette, la maggior parte di esse, devono trovarsi nella sottodirectory. `recipes`

È ora possibile aggiungere la ricetta al libro di ricette e utilizzare Test Kitchen per eseguirla su un'istanza.

Per eseguire la ricetta

1. Creare un file denominato `default.rb` che contenga il codice di esempio dell'installazione Git tratto dall'inizio della sezione e salvarlo nella sottodirectory `recipes`.
2. Nella directory `installpkg`, eseguire `kitchen converge`. Questo comando avvia una nuova istanza di Ubuntu in Vagrant, copia i tuoi libri di cucina nell'istanza e avvia un'esecuzione di Chef per eseguire le ricette nell'elenco di esecuzione. `.kitchen.yml`

3. Per verificare che la ricetta sia riuscita, eseguire `kitchen login`, che apre una connessione SSH all'istanza. Quindi eseguire `git --version` per verificare la corretta installazione di Git. Per tornare alla workstation, eseguire `exit`.
4. Al termine dell'operazione, eseguire `kitchen destroy` per chiudere l'istanza. L'esempio successivo utilizza un altro libro di ricette.

Questo esempio è un buon inizio, ma è molto semplice. L'installazione di altri pacchetti può risultare più complicata e richiedere l'esecuzione di una o più delle seguenti operazioni:

- Creare e configurare un utente.
- Creare una o più directory per dati, log e così via.
- Installare uno o più file di configurazione.
- Specificare nomi del pacchetto o valori degli attributi diversi per differenti sistemi operativi.
- Avviare un servizio e quindi riavviarlo in base alle esigenze.

Gli esempi seguenti descrivono come agire in questi casi e alcune altre utili operazioni.

Esempio 2: gestione degli utenti

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Un'altra semplice attività è la gestione degli utenti su un'istanza. La seguente ricetta aggiunge un nuovo utente a un'istanza di Linux.

```
user "myuser" do
  home "/home/newuser"
  shell "/bin/bash"
end
```

È possibile utilizzare una risorsa utente per gestire gli utenti sia su sistemi Linux che Windows, anche se alcuni attributi si applicano a un solo sistema. L'esempio crea un utente denominato `myuser` e specifica directory principale e shell personali. Non vi sono operazioni specificate, perciò la risorsa utilizza l'operazione di default `create`. È possibile aggiungere attributi a `user` per specificare una serie di altre impostazioni, ad esempio la password o l'ID gruppo. È inoltre possibile utilizzare `user` per operazioni correlate alle gestione degli utenti, ad esempio per modificare le impostazioni utente o eliminare gli utenti. Per ulteriori informazioni, consulta la sezione relativa alla risorsa user.

Per eseguire la ricetta

1. Creare una directory in `opsworks_cookbooks` denominata `newuser` e accedervi.
2. Creare un file `metadata.rb` che contiene il codice seguente e salvarlo in `newuser`.

   ```
   name "newuser"
   version "0.1.0"
   ```

3. Inizializzare e configurare Test Kitchen come descritto in Esempio 1: installazione di pacchetti, quindi aggiungere una directory `recipes` all'interno della directory `newuser`.
4. Aggiungere il file `default.rb` con la ricetta di esempio alla directory `recipes` del libro di ricette.
5. Eseguire `kitchen converge` per eseguire la ricetta.
6. Usare `kitchen login` per accedere all'istanza e verificare l'esistenza del nuovo utente eseguendo `cat /etc/passwd`. L'utente `myuser` deve trovarsi in fondo al file.

Esempio 3: creazione di directory

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Quando installi un pacchetto su un'istanza, spesso devi creare alcuni file di configurazione e inserirli nella directory appropriata. Tuttavia, tali directory potrebbero non esistere ancora. È inoltre possibile

che sia necessario creare directory per dati, file di log e così via. Ad esempio, per prima cosa avvii il sistema Ubuntu che usi per la maggior parte degli esempi, la `/srv` directory non ha sottodirectory. Se stai installando un server delle applicazioni, è probabile che tu voglia una directory `/srv/www/` e forse alcune sottodirectory per file di dati, log e così via. La ricetta seguente crea `/srv/www/` su un'istanza.

```
directory "/srv/www/" do
  mode 0755
  owner 'root'
  group 'root'
  action :create
end
```

Una risorsa `directory` serve a creare e configurare directory su sistemi Linux e Windows, anche se alcuni attributi vengono utilizzati in modo diverso. Il nome della risorsa è il valore di default per l'attributo `path` della risorsa, perciò l'esempio crea `/srv/www/` e ne specifica le proprietà `mode`, `owner` e `group`.

Per eseguire la ricetta

1. Creare una directory in `opsworks_cookbooks` denominata `createdir` e aprirla.
2. Inizializzare e configurare Test Kitchen come descritto in Esempio 1: installazione di pacchetti, quindi aggiungere una directory `recipes` all'interno di `createdir`.
3. Aggiungere un file `default.rb` con il codice della ricetta alla sottodirectory `recipes` del libro di ricette.
4. Eseguire `kitchen converge` per eseguire la ricetta.
5. Eseguire `kitchen login`, individuare `/srv` e verificare che includa una sottodirectory `www`.
6. Eseguire `exit` per tornare alla workstation, ma lasciare l'istanza in esecuzione.

> **Note**
>
> Per creare una directory relativa alla directory principale dell'istanza, utilizzare `#{ENV['HOME']}` per rappresentare la directory principale. Ad esempio, quanto segue crea la directory `~/shared`.

```
directory "#{ENV['HOME']}/shared" do
  ...
end
```

Supponiamo di voler creare una directory nidificata più profondamente, ad esempio `/srv/www/shared`. È possibile modificare la ricetta precedente come segue.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  action :create
end
```

Per eseguire la ricetta

1. Sostituire il codice in `default.rb` con la ricetta precedente.
2. Eseguire `kitchen converge` nella directory `createdir`.
3. Per verificare che la directory sia stata effettivamente creata, eseguire `kitchen login`, individuare `/srv/www` e verificare che contenga una sottodirectory `shared`.
4. Eseguire `kitchen destroy` per arrestare l'istanza.

Noterai che il comando `kitchen converge` è stato eseguito molto più velocemente. Questo perché l'istanza è già in esecuzione, quindi non è necessario avviare l'istanza, installare Chef e così via. Test Kitchen si limita a copiare il libro di ricette aggiornato nell'istanza e avvia un'esecuzione Chef.

Ora esegui nuovamente `kitchen converge`, che esegue la ricetta su una nuova istanza. Verrà visualizzato un risultato analogo al seguente.

```
Chef Client failed. 0 resources updated in 1.908125788 seconds
[2014-06-20T20:54:26+00:00] ERROR: directory[/srv/www/shared] (createdir::default line
 1) had an error: Chef::Exceptions::EnclosingDirectoryDoesNotExist: Parent directory /
srv/www does not exist, cannot create /srv/www/shared
[2014-06-20T20:54:26+00:00] FATAL: Chef::Exceptions::ChildConvergeError: Chef run
 process exited unsuccessfully (exit code 1)
>>>>>> Converge failed on instance <default-ubuntu-1204>.
```

```
>>>>>> Please see .kitchen/logs/default-ubuntu-1204.log for more details
>>>>>> ------Exception-------
>>>>>> Class: Kitchen::ActionFailed
>>>>>> Message: SSH exited (1) for command: [sudo -E chef-solo --config /tmp/kitchen/
solo.rb --json-attributes /tmp/kitchen/dna.json  --log_level info]
>>>>>> ----------------------
```

Che cos'è successo? Il problema è che, per impostazione predefinita, una risorsa `directory` può creare una sola directory alla volta, non una serie di directory. La ricetta ha funzionato in precedenza perché la prima ricetta eseguita sull'istanza aveva già creato `/srv/www`, perciò con la creazione di `/srv/www/shared` è stata creata una sola sottodirectory.

> **Note**
>
> Quando esegui `kitchen converge`, verifica se stai eseguendo le ricette su un'istanza nuova o esistente, perché potresti ottenere risultati differenti.

Per creare una serie di sottodirectory, aggiungi un attributo `recursive` a `directory` e impostalo su `true`. La ricetta seguente crea `/srv/www/shared` direttamente su un'istanza pulita.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

Esempio 4: aggiunta del controllo del flusso

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Alcune ricette sono solo una serie di risorse Chef. In questo caso, all'esecuzione, la ricetta esegue semplicemente ciascuno dei provider di risorse in sequenza. Tuttavia, spesso è utile avere un percorso di esecuzione più sofisticato. Di seguito sono elencati due scenari comuni:

- Una ricetta deve eseguire la stessa risorsa più volte con diverse impostazioni di attributo.
- Vanno utilizzate diverse impostazioni di attributo su differenti sistemi operativi.

È possibile affrontare situazioni come queste incorporando strutture di controllo di Ruby nella ricetta. Questa sezione spiega come modificare la ricetta da Esempio 3: creazione di directory per affrontare entrambi gli scenari.

Argomenti

- Iterazione
- Logica condizionale

Iterazione

In Esempio 3: creazione di directory è stato illustrato come utilizzare una risorsa `directory` per creare una directory o serie di directory. Tuttavia, supponiamo di voler creare due diverse directory, `/srv/www/config` e `/srv/www/shared`. È possibile implementare una risorsa di directory separata per ciascuna directory, ma questo approccio può rivelarsi complesso se intendi creare molte directory. La seguente ricetta mostra un modo più semplice di gestire l'attività.

```
[ "/srv/www/config", "/srv/www/shared" ].each do |path|
  directory path do
    mode 0755
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
end
```

Invece di utilizzare una risorsa di directory separata per ciascuna sottodirectory, la ricetta utilizza una raccolta di stringa che contiene i percorsi delle sottodirectory. Il metodo `each` di Ruby esegue la risorsa una volta per ciascun elemento della raccolta, a partire dal primo. Il valore dell'elemento è rappresentato nella risorsa dalla variabile `path`, che in questo caso rappresenta il percorso di

directory. È possibile adattare questo esempio con semplicità per creare un qualsiasi numero di sottodirectory.

Per eseguire la ricetta

1. Rimanere nella directory `createdir`; tale libro di ricette sarà utilizzato per i prossimi esempi.
2. Se non è già stato fatto, avviare `kitchen destroy` per iniziare con un'istanza pulita.
3. Sostituire il codice in `default.rb` con l'esempio ed eseguire `kitchen converge`.
4. Accedere all'istanza; la directory appena creata sarà disponibile in `/srv`.

È possibile utilizzare una tabella hash per specificare due valori per ogni iterazione. La seguente ricetta crea `/srv/www/config` e `/srv/www/shared`, ognuna con una modalità diversa.

```
{ "/srv/www/config" => 0644, "/srv/www/shared" => 0755 }.each do |path, mode_value|
  directory path do
    mode mode_value
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
end
```

Per eseguire la ricetta

1. Se non è già stato fatto, avviare `kitchen destroy` per iniziare con un'istanza pulita.
2. Sostituire il codice in `default.rb` con l'esempio ed eseguire `kitchen converge`.
3. Accedere all'istanza; la directory appena creata sarà disponibile in `/srv` con le modalità specificate.

> **Note**
>
> AWS OpsWorks Le ricette Stacks utilizzano comunemente questo approccio per estrarre valori dalla configurazione dello stack e dalla distribuzione JSON, che è fondamentalmente una grande tabella hash, e inserirli in una risorsa. Per vedere un esempio, consulta Ricette di ditribuzione.

Logica condizionale

È anche possibile usare la logica condizionale di Ruby per creare più rami di esecuzione. La seguente ricetta utilizza la logica `if-elsif-else` per estendere l'esempio precedente in modo da creare una sottodirectory denominata `/srv/www/shared`, ma solo su sistemi Ubuntu e Debian. Per tutti gli altri sistemi, registra un messaggio di errore che viene visualizzato nell'output di Test Kitchen.

```
if platform?("debian", "ubuntu")
  directory "/srv/www/shared" do
    mode 0755
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
else
  log "Unsupported system"
end
```

Per eseguire la ricetta di esempio

1. Se l'istanza è ancora attiva, eseguire `kitchen destroy` per chiuderla.
2. Sostituire il codice in `default.rb` con il codice di esempio.
3. Modificare `.kitchen.yml` per aggiungere un sistema CentOS 6.4 all'elenco delle piattaforme. La sezione `platforms` del file avrà adesso un aspetto simile al seguente.

   ```
   ...
   platforms:
     - name: ubuntu-12.04
     - name: centos-6.4
   ...
   ```

4. Eseguire `kitchen converge`, che consente di creare un'istanza ed eseguire le ricette per ogni piattaforma in `.kitchen.yml`, in sequenza.

> **Note**
>
> Per convergere una sola istanza, aggiungere il nome dell'istanza come parametro. Ad esempio, per convergere la ricetta solo sulla piattaforma Ubuntu, eseguire `kitchen converge default-ubuntu-1204`. In caso di dimenticanza dei nomi della piattaforma, basta eseguire `kitchen list`.

Il messaggio di log dovrebbe essere visualizzato nella parte riservata a CentOS dell'output di Test Kitchen, che sarà simile alla seguente:

```
...
Converging 1 resources
Recipe: createdir::default
* log[Unsupported system] action write[2014-06-23T19:10:30+00:00] INFO: Processing
 log[Unsupported system] action write (createdir::default line 12)
[2014-06-23T19:10:30+00:00] INFO: Unsupported system

[2014-06-23T19:10:30+00:00] INFO: Chef Run complete in 0.004972162 seconds
```

Ora puoi accedere alle istanze e verificare se le directory sono state create o meno. Tuttavia, non puoi eseguire semplicemente `kitchen login` ora. Devi specificare l'istanza aggiungendo il nome della piattaforma, ad esempio `kitchen login default-ubuntu-1204` .

> **Note**
>
> Se un comando Test Kitchen richiede un nome di istanza, non è necessario digitare il nome completo. Test Kitchen considera il nome di un'istanza come espressione regolare di Ruby, perciò serve solo un numero di caratteri sufficiente a fornire una corrispondenza univoca. Ad esempio, è possibile convergere la sola istanza Ubuntu eseguendo `kitchen converge ub` o accedendo all'istanza CentOS eseguendo `kitchen login 64`.

A questo punto probabilmente ti starai chiedendo come fa la ricetta a sapere su quale piattaforma è in esecuzione. Per ogni sessione, Chef esegue uno strumento chiamato Ohai che raccoglie i dati di sistema, tra cui la piattaforma, e li rappresenta come una serie di attributi in una struttura chiamata

nodo oggetto. Il metodo `platform?` di Chef confronta i sistemi tra parentesi con il valore della piattaforma Ohai e restituisce true in caso di corrispondenza.

È possibile fare riferimento al valore di un attributo di nodo direttamente nel codice utilizzando `node['attribute_name']`. Il valore della piattaforma, ad esempio, è rappresentato da `node['platform']`. È possibile, ad esempio, aver scritto l'esempio precedente come segue.

```
if node[:platform] == 'debian' or node[:platform] == 'ubuntu'
  directory "/srv/www/shared" do
    mode 0755
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
else
  log "Unsupported system"
end
```

Un motivo comune per includere una logica condizionale in una ricetta si basa sul fatto che diverse famiglie Linux talvolta utilizzano nomi diversi per pacchetti, directory e così via. Ad esempio, il nome del pacchetto Apache è `httpd` sui sistemi CentOS e `apache2` sui sistemi Ubuntu.

Se serve solo un'altra stringa per diversi sistemi, il metodo `value_for_platform` di Chef è una soluzione più semplice rispetto a `if-elsif-else`. La seguente ricetta crea una directory `/srv/www/shared` su sistemi CentOS, una directory `/srv/www/data` su sistemi Ubuntu e `/srv/www/config` su tutti gli altri.

```
data_dir = value_for_platform(
  "centos" => { "default" => "/srv/www/shared" },
  "ubuntu" => { "default" => "/srv/www/data" },
  "default" => "/srv/www/config"
)
directory data_dir do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
```

```
end
```

`value_for_platform` assegna il percorso appropriato a `data_dir` e la risorsa `directory` utilizza tale valore per creare la directory.

Per eseguire la ricetta di esempio

1. Se l'istanza è ancora attiva, eseguire `kitchen destroy` per chiuderla.
2. Sostituire il codice in `default.rb` con il codice di esempio.
3. Esegui `kitchen converge` e quindi accedere a ogni istanza per verificare che siano presenti le directory appropriate.

Esempio 5: utilizzo di attributi

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le ricette nelle sezioni precedenti utilizzavano valori prefissati per tutto, tranne che per la piattaforma. Questo approccio può rivelarsi poco pratico se, ad esempio, vuoi utilizzare lo stesso valore in più ricette. Puoi definire i valori separatamente dalle ricette includendo un file attributo nel tuo libro di ricette.

Un file attributo è un'applicazione Ruby che assegna valori a uno o più attributi. deve trovarsi nella cartella `attributes` del libro di ricette. Chef incorpora gli attributi nell'oggetto nodo e qualsiasi ricetta può utilizzare i valori dell'attributo facendovi riferimento. Questo argomento spiega come modificare la ricetta da Iterazione per utilizzare gli attributi. Ecco la ricetta originale come riferimento.

```
[ "/srv/www/config", "/srv/www/shared" ].each do |path|
  directory path do
    mode 0755
    owner 'root'
```

```
    group 'root'
    recursive true
    action :create
  end
end
```

Di seguito vengono determinati gli attributi per il nome della sottodirectory, la modalità, il proprietario e i valori di gruppo.

```
default['createdir']['shared_dir'] = 'shared'
default['createdir']['config_dir'] = 'config'
default['createdir']['mode'] = 0755
default['createdir']['owner'] = 'root'
default['createdir']['group'] = 'root'
```

Tieni presente quanto segue:

- Ogni definizione inizia con un tipo di attributo.

  Se un attributo viene definito più di una volta, magari in file di attributi diversi, il tipo di attributo specifica la precedenza dell'attributo, che determina quale definizione è incorporata nell'oggetto nodo. Per ulteriori informazioni, consulta Precedenza degli attributi. Tutte le definizioni in questo esempio hanno il tipo di attributo `default`, che è l'opzione più comune per questo scopo.
- Gli attributi hanno nomi nidificati.

  L'oggetto nodo è fondamentalmente una tabella hash che può essere nidificata con profondità arbitraria, perciò i nomi di attributo possono essere nidificati ed è comune che lo siano. Questo file attributo segue una pratica standard, ovvero l'utilizzo di un nome nidificato con il nome del libro di ricette, `createdir`, come primo elemento.

L'utilizzo di createdir come primo elemento dell'attributo è dovuto al fatto che Chef, durante l'esecuzione di una sessione, incorpora gli attributi di ogni libro di ricette nell'oggetto nodo. Con AWS OpsWorks Stacks, l'oggetto node include un gran numero di attributi dei libri di cucina incorporati oltre a tutti gli attributi definiti dall'utente. Includere il nome del libro di ricette nel nome dell'attributo riduce il rischio di un conflitto di nomi con gli attributi di un altro libro di ricette, soprattutto se l'attributo ha un nome come `port` o `user`. Non assegnare a un attributo un nome del tipo `[:apache2][:user]`, ad esempio, a meno di voler sovrascrivere il valore di tale attributo. Per ulteriori informazioni, consulta Utilizzo degli attributi personalizzati del libro di ricette.

L'esempio seguente mostra la ricetta originale con gli attributi invece che con i valori prefissati.

```
[ "/srv/www/#{node['createdir']['shared_dir']}", "/srv/www/#{node['createdir']
['config_dir']}" ].each do |path|
  directory path do
    mode node['createdir']['mode']
    owner node['createdir']['owner']
    group node['createdir']['group']
    recursive true
    action :create
  end
end
```

> **Note**
>
> Se vuoi incorporare un valore attributo in una stringa, includilo fra `#{}`. Nell'esempio precedente, `#{node['createdir']['shared_dir']}` aggiunge "shared" a "/srv/www/".

Per eseguire la ricetta

1. Eseguire `kitchen destroy` per iniziare con un'istanza pulita.
2. Sostituire il codice in `recipes/default.rb` con l'esempio della ricetta precedente.
3. Creare una sottodirectory di `createdir` denominata `attributes` e aggiungere un file con nome `default.rb` che contiene le definizioni degli attributi.
4. Modificare `.kitchen.yml` per rimuovere CentOS dall'elenco delle piattaforme.
5. Eseguire `kitchen converge`, effettuare l'accesso all'istanza e verificare la presenza di `/srv/www/shared` e `/srv/www/config`.

> **Note**
>
> Con AWS OpsWorks Stacks, la definizione dei valori come attributi offre un ulteriore vantaggio; puoi utilizzare JSON personalizzato per sovrascrivere tali valori in base allo stack o anche alla distribuzione. Questa possibilità può essere utile per molti scopi, ad esempio:
>
> - È possibile personalizzare il comportamento delle ricette, ad esempio le impostazioni di configurazione o i nomi utente, senza dover modificare il libro di ricette.

È possibile, ad esempio, utilizzare lo stesso libro di ricette per diversi stack e utilizzare un JSON personalizzato per specificare le impostazioni di configurazione chiave per un determinato stack. Questo consente di risparmiare il tempo e il lavoro necessari per modificare il libro di ricette o utilizzare un diverso libro di ricette per ogni stack.

- Non inserire informazioni potenzialmente riservate, ad esempio le password del database, nel repository del libro di ricette.

  Puoi utilizzare un attributo per definire un valore di default e quindi utilizzare JSON personalizzati per sovrascrivere quel valore con quello reale.

Per ulteriori informazioni sull'utilizzo di JSON personalizzati per sovrascrivere gli attributi, consulta Sostituzione degli attributi.

Il file attributo è denominato `default.rb` perché è un'applicazione Ruby, anche se semplice. Ciò significa che è possibile, ad esempio, utilizzare la logica condizionale per specificare valori degli attributi in base al sistema operativo. In Logica condizionale, hai specificato nella ricetta un nome di sottodirectory diverso per differenti famiglie Linux. Un file attributo, invece, ti consente di inserire la logica condizionale nel file attributo stesso.

Il seguenti file attributo impiega `value_for_platform` per specificare un valore di attributo `['shared_dir']` diverso a seconda del sistema operativo. Per altre condizioni, puoi usare la logica `if-elsif-else` di Ruby o un'istruzione `case`.

```
data_dir = value_for_platform(
  "centos" => { "default" => "shared" },
  "ubuntu" => { "default" => "data" },
  "default" => "user_data"
)
default['createdir']['shared_dir'] = data_dir
default['createdir']['config_dir'] = "config"
default['createdir']['mode'] = 0755
default['createdir']['owner'] = 'root'
default['createdir']['group'] = 'root'
```

Per eseguire la ricetta

1. Eseguire `kitchen destroy` per iniziare con un'istanza nuova.
2. Sostituire il codice in `attributes/default.rb` con l'esempio precedente.
3. Modificare `.kitchen.yml` per aggiungere una piattaforma CentOS alla sezione delle piattaforme, come descritto in Logica condizionale.
4. Eseguire `kitchen converge` e quindi effettuare l'accesso alle istanze per verificare la presenza delle directory.

Al termine dell'operazione, eseguire `kitchen destroy` per terminare l'istanza. L'esempio successivo utilizza un nuovo libro di ricette.

Esempio 6: creazione di file

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver creato le directory, spesso è necessario aggiungervi file di configurazione, file di dati e così via. Questo argomento mostra due modi per installare i file su un'istanza.

Argomenti

- Installazione di un file da un libro di ricette
- Creazione di un file da un modello

Installazione di un file da un libro di ricette

Il modo più semplice di installare un file su un'istanza è utilizzando una risorsa `cookbook_file`, che copia un file dal libro di ricette a un percorso specificato sull'istanza, per sistemi sia Linux che Windows. Questo esempio estende la ricetta da Esempio 3: creazione di directory per aggiungere un file di dati `/srv/www/shared` dopo la creazione della directory. Ecco la ricetta originale come riferimento.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

Per configurare il libro di ricette

1. Nella directory `opsworks_cookbooks`, creare una directory denominata `createfile` e aprirla.
2. Aggiungere un file `metadata.rb` a `createfile` con il contenuto seguente.

   ```
   name "createfile"
   version "0.1.0"
   ```

3. Inizializzare e configurare Test Kitchen come descritto in Esempio 1: installazione di pacchetti, quindi rimuovere CentOS dall'elenco `platforms`.
4. Aggiungere la sottodirectory `recipes` a `createfile`.

Il file da installare contiene i seguenti dati JSON.

```
{
  "my_name" : "myname",
  "your_name" : "yourname",
  "a_number" : 42,
  "a_boolean" : true
}
```

Per configurare il file di dati

1. Aggiungere la sottodirectory `files` a `createfile` e la sottodirectory `default` a `files`. Qualsiasi file installato con `cookbook_file` deve trovarsi in una sottodirectory di `files`, come `files/default` in questo esempio.

> **Note**
>
> Per specificare diversi file per sistemi differenti, è possibile inserire ogni file specifico in una sottocartella denominata secondo il sistema, ad esempio `files/ubuntu`. La risorsa `cookbook_file` copia l'appropriato file specifico per il sistema, se esistente, altrimenti utilizza il file `default`. Per ulteriori informazioni, consulta cookbook_file.

2. Creare un file denominato `example_data.json` con il JSON dell'esempio precedente e aggiungerlo a `files/default`.

La seguente ricetta copia `example_data.json` in un percorso specificato.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end

cookbook_file "/srv/www/shared/example_data.json" do
  source "example_data.json"
  mode 0644
  action :create_if_missing
end
```

Dopo che la risorsa della directory crea `/srv/www/shared`, la risorsa `cookbook_file` copia `example_data.json` in tale directory e imposta anche l'utente, il gruppo e la modalità del file.

> **Note**
>
> La risorsa `cookbook_file` introduce una nuova operazione: `create_if_missing`. È inoltre possibile utilizzare un'operazione `create`, che però sovrascrive un file esistente. Se non si desidera sovrascrivere nulla, utilizzare `create_if_missing`, che installa `example_data.json` solo se non è già esistente.

Per eseguire la ricetta

1. Eseguire `kitchen destroy` per iniziare con un'istanza nuova.
2. Creare un file `default.rb` che contiene la ricetta precedente e salvarlo in `recipes`.
3. Eseguire `kitchen converge`, quindi accedere all'istanza per verificare che `/srv/www/shared` contenga `example_data.json`.

Creazione di un file da un modello

La risorsa `cookbook_file` è utile per alcuni scopi, ma si limita a installare qualsiasi file presente nel libro di ricette. Una risorsa `template` offre un modo più flessibile di installare un file su un'istanza di Windows o Linux tramite creazione dinamica da un modello. È quindi possibile determinare i dettagli del contenuto del file in fase di runtime e apportare modifiche in base alle esigenze. Ad esempio, potresti volere che un file di configurazione abbia una determinata impostazione all'avvio dell'istanza per poi modificare l'impostazione in seguito all'aggiunta di altre istanze allo stack.

Questo esempio modifica il libro di ricette `createfile` perché sia utilizzata una risorsa `template` per installare una versione di `example_data.json` leggermente modificata.

Ecco come apparirà il file installato.

```
{
  "my_name" : "myname",
  "your_name" : "yourname",
  "a_number" : 42,
  "a_boolean" : true,
  "a_string" : "some string",
  "platform" : "ubuntu"
}
```

Le risorse del modello sono in genere utilizzate in combinazione con i file attributo, quindi l'esempio ne utilizza uno per definire i seguenti valori.

```
default['createfile']['my_name'] = 'myname'
default['createfile']['your_name'] = 'yourname'
default['createfile']['install_file'] = true
```

Per configurare il libro di ricette

1. Eliminare la directory `createfile` del libro di ricette `files` e il relativo contenuto.
2. Aggiungere la sottodirectory `attributes` a `createfile` e aggiungere a `default.rb` il file `attributes` che contiene le definizioni degli attributi precedenti.

Un modello è un file `.erb` che è fondamentalmente una copia del file finale, con alcuni dei contenuti rappresentati da segnaposti. Quando la risorsa `template` crea il file, copia i contenuti del modello nel file specificato e sovrascrive i segnaposto con i relativi valori assegnati. Ecco il modello per `example_data.json`.

```
{
  "my_name" : "<%= node['createfile']['my_name'] %>",
  "your_name" : "<%= node['createfile']['your_name'] %>",
  "a_number" : 42,
  "a_boolean" : <%= @a_boolean_var %>,
  "a_string" : "<%= @a_string_var %>",
  "platform" : "<%= node['platform'] %>"
}
```

I valori `<%=...%>` sono i segnaposto.

- `<%=node[...]%>` rappresenta un nodo valore attributo.

  In questo esempio, il valore "your_name" è un segnaposto che rappresenta uno dei valori attributo del file attributo del libro di ricette.
- `<%=@...%>` rappresenta il valore di una variabile definita nella risorsa del modello, come illustreremo più avanti.

Per creare il file del modello

1. Aggiungere la sottodirectory `templates` al libro di ricette `createfile` e la sottodirectory `default` a `templates`.

   > **Note**
   >
   > La directory `templates` funziona in modo analogo alla directory `files`. È possibile inserire modelli specifici per il sistema in una sottodirectory, ad esempio `ubuntu`, con

nome assegnato a seconda del sistema. La risorsa `template` usa l'appropriato modello specifico per il sistema, se esistente, altrimenti utilizza il modello `default`.

2. Creare un file denominato `example_data.json.erb` e salvarlo nella directory `templates/default`. Il nome del modello è arbitrario, ma è in genere si crea aggiungendo `.erb` al nome del file, incluse le estensioni.

La seguente ricetta utilizza una risorsa `template` per creare `/srv/www/shared/example_data.json`.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end

template "/srv/www/shared/example_data.json" do
  source "example_data.json.erb"
  mode 0644
  variables(
    :a_boolean_var => true,
    :a_string_var => "some string"
  )
  only_if {node['createfile']['install_file']}
end
```

La risorsa `template` crea `example_data.json` da un modello e lo installa in `/srv/www/shared`.

- Il nome del modello, `/srv/www/shared/example_data.json`, specifica il percorso e il nome del file installato.
- L'attributo `source` specifica il modello utilizzato per creare il file.
- L'attributo `mode` specifica la modalità di installazione del file.
- La risorsa definisce due variabili: `a_boolean_var` e `a_string_var`.

  Quando la risorsa crea `example_data.json`, sovrascrive la variabile segnaposto del modello con i valori corrispondenti della risorsa.

- L'attributo di `only_if`protezione indica alla risorsa di creare il file solo se `['createfile'] ['install_file']` è impostato su `true`.

Per eseguire la ricetta

1. Eseguire `kitchen destroy` per iniziare con un'istanza nuova.
2. Sostituire il codice in `recipes/default.rb` con l'esempio precedente.
3. Eseguire `kitchen converge`, quindi effettuare l'accesso all'istanza per verificare che il file sia presente in `/srv/www/shared` con i contenuti corretti.

Al termine dell'operazione, eseguire `kitchen destroy` per chiudere l'istanza. La sezione successiva utilizza un nuovo libro di ricette.

Esempio 7: esecuzione di comandi e script

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le risorse di Chef possono gestire un'ampia gamma di attività su un'istanza, ma talvolta è preferibile usare un comando shell o uno script. Ad esempio, se hai già script che utilizzi per eseguire determinate attività, sarà più semplice continuare a utilizzar questi ultimi, piuttosto che implementare un nuovo codice. Questa sezione illustra come eseguire comandi o script in un'istanza.

Argomenti

- Esecuzione di comandi
- Esecuzione di script

Esecuzione di comandi

La risorsa `script` esegue uno o più comandi. Supporta gli interpreti di comandi csh, bash, Perl, Python e Ruby, quindi può essere utilizzato sia su sistemi Linux che Windows, purché vi sia installato

l'opportuno interprete. Questo argomento illustra come eseguire un semplice comando bash su un'istanza Linux. Chef supporta anche le risorse powershell_script e batch per l'esecuzione di script su Windows. Per ulteriori informazioni, consulta Esecuzione di uno PowerShell script di Windows.

Per iniziare

1. Nella directory `opsworks_cookbooks`, creare una directory denominata `script` e aprirla.
2. Aggiungere un file `metadata.rb` a `script` con il contenuto seguente.

```
name "script"
version "0.1.0"
```

3. Inizializzare e configurare Test Kitchen come descritto in Esempio 1: installazione di pacchetti, quindi rimuovere CentOS dall'elenco `platforms`.
4. In `script`, creare una directory denominata `recipes`.

È possibile eseguire comandi utilizzando la stessa risorsa `script`, ma Chef supporta anche un set di versioni della risorsa specifico per interprete di comandi; tali versioni vengono denominate a seconda dell'interprete. La seguente ricetta utilizza una risorsa `bash` per eseguire un semplice script bash.

```
bash "install_something" do
  user "root"
  cwd "/tmp"
  code <<-EOH
    touch somefile
  EOH
  not_if do
    File.exists?("/tmp/somefile")
  end
end
```

La risorsa `bash` è configurata come segue.

- Utilizza l'operazione di default, `run`, che esegue i comandi nel blocco `code`.

  Questo esempio contiene un solo comando, `touch somefile`ma un blocco `code` può contenere più comandi.
- L'attributo `user` specifica l'utente che esegue il comando.

- L'attributo `cwd` specifica la directory di lavoro.

  Per questo esempio, `touch` crea un file nella directory `/tmp`.
- L'attributo di protezione `not_if` indica alla risorsa di non eseguire alcuna operazione se il file è già esistente.

Per eseguire la ricetta

1. Creare un file `default.rb` che contiene il codice di esempio precedente e salvarlo in `recipes`.
2. Eseguire `kitchen converge`, quindi accedere all'istanza per verificare che il file sia presente in `/tmp`.

Esecuzione di script

La risorsa `script` è pratica, soprattutto se hai bisogno di eseguire solo uno o due comandi; spesso è però preferibile archiviare lo script in un file ed eseguire il file stesso. La risorsa `execute` esegue uno specifico file eseguibile, inclusi i file di script, in Linux o Windows. Questo argomento modifica il libro di ricette `script` dell'esempio precedente per usare `execute` al fine di eseguire un semplice script shell. È possibile estendere facilmente l'esempio a script più complessi o ad altri tipi di file eseguibile.

Per configurare il file script

1. Aggiungere la sottodirectory `files` a `script` e la sottodirectory `default` a `files`.
2. Creare un file denominato `touchfile` che contenga quando segue e aggiungerlo a `files/default`. In questo esempio viene utilizzata una comune riga di interpretazione Bash, ma puoi sostituire l'interprete con quello più adatto al tuo ambiente shell, se necessario.

   ```
   #!/usr/bin/env bash
   touch somefile
   ```

   Il file script può contenere un numero qualsiasi di comandi. Per comodità, questo script di esempio include un solo comando `touch`.

La seguente ricetta esegue lo script.

```
cookbook_file "/tmp/touchfile" do
```

```
  source "touchfile"
  mode 0755
end

execute "touchfile" do
  user "root"
  cwd "/tmp"
  command "./touchfile"
end
```

La risorsa `cookbook_file` copia il file script su `/tmp` e imposta la modalità per rendere il file eseguibile. La risorsa `execute` esegue quindi il file come indicato di seguito:

- L'attributo `user` specifica l'utente del comando (`root` in questo esempio).
- L'attributo `cwd` specifica la directory di lavoro (`/tmp` in questo esempio).
- L'attributo `command` specifica lo script da eseguire (`touchfile` in questo esempio), che si trova nella directory di lavoro.

Per eseguire la ricetta

1. Sostituire il codice in `recipes/default.rb` con l'esempio precedente.
2. Eseguire `kitchen converge`, quindi effettuare l'accesso all'istanza per verificare che `/tmp` ora contenga il file script, con la modalità impostata su 0755, e `somefile`.

Al termine dell'operazione, eseguire `kitchen destroy` per chiudere l'istanza. La sezione successiva utilizza un nuovo libro di ricette.

Esempio 8: gestione dei servizi

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

I pacchetti, ad esempio i server delle applicazioni, in genere hanno associato un servizio che deve essere avviato, arrestato, riavviato e così via. Ad esempio, è necessario avviare il servizio Tomcat dopo l'installazione del pacchetto o dopo il termine del processo di avvio dell'istanza e riavviare il servizio a ogni modifica del file di configurazione. In questo argomento vengono illustrati i concetti di base per la gestione di un servizio su un'istanza Linux, utilizzando un server delle applicazioni Tomcat come esempio. La risorsa service funziona quasi nello stesso modo sulle istanze Windows, anche se vi sono alcune differenze nei dettagli. Per ulteriori informazioni, consulta `service`.

> **Note**
>
> L'esempio prevede un'installazione minima di Tomcat, quanto basta per illustrare le procedure fondamentali per l'utilizzo di una risorsa `service`. Per un esempio di come implementare ricette per un server Tomcat più funzionale, consulta Creazione di un livello server Tomcat personalizzato.

Argomenti

- Definizione e avvio di un servizio
- Utilizzo di notifies per avviare o riavviare un servizio

Definizione e avvio di un servizio

Questa sezione mostra i concetti di base per definire e avviare un servizio.

Per iniziare

1. Nella directory `opsworks_cookbooks`, creare una directory denominata `tomcat` e aprirla.
2. Aggiungere un file `metadata.rb` a `tomcat` con il contenuto seguente.

   ```
   name "tomcat"
   version "0.1.0"
   ```

3. Inizializzare e configurare Test Kitchen come descritto in Esempio 1: installazione di pacchetti, quindi rimuovere CentOS dall'elenco `platforms`.
4. Aggiungere la sottodirectory `recipes` a `tomcat`.

La gestione del servizio avviene tramite una risorsa `service`. La seguente ricetta di default installa Tomcat e avvia il servizio.

```
execute "install_updates" do
  command "apt-get update"
end

package "tomcat7" do
    action :install
end

include_recipe 'tomcat::service'

service 'tomcat' do
  action :start
end
```

La ricetta effettua le seguenti operazioni:

- La risorsa `execute` esegue `apt-get update` per installare gli aggiornamenti correnti del sistema.

  Per l'istanza di Ubuntu utilizzata in questo esempio, è necessario installare gli aggiornamenti prima di installare Tomcat. Altri sistemi possono avere requisiti diversi.
- La risorsa `package` installa Tomcat 7.
- La ricetta `tomcat::service` inclusa definisce il servizio e verrà discussa più avanti.
- La risorsa `service` avvia il servizio Tomcat.

  È anche possibile usare questa risorsa per l'esecuzione di altri comandi, ad esempio per interrompere e riavviare il servizio.

L'esempio seguente illustra la ricetta `tomcat::service`.

```
service 'tomcat' do
  service_name "tomcat7"
  supports :restart => true, :reload => false, :status => true
  action :nothing
end
```

Questa ricetta crea la definizione del servizio Tomcat come segue:

- Il nome della risorsa, `tomcat`, viene utilizzato da altre ricette per fare riferimento al servizio.

  Ad esempio, `default.rb` fa riferimento a `tomcat` per avviare il servizio.
- La risorsa `service_name` specifica il nome del servizio.

  Nell'elenco dei servizi per l'istanza, il servizio Tomcat sarà denominato tomcat7.
- `supports` specifica come Chef gestisce i comandi `restart`, `reload` e `status` del servizio.
  - `true` indica che Chef può utilizzare lo script init o un altro provider di servizi per eseguire il comando.
  - `false` indica che Chef deve tentare di eseguire il comando con altri mezzi.

Osserva che `action` è impostato su `:nothing`, che indica alla risorsa di non eseguire alcuna operazione. La risorsa service supporta azioni quali `start` e `restart`. Tuttavia, questo libro di ricette segue una pratica standard: l'utilizzo della definizione di un servizio che non richiede alcun intervento con avvio o riavvio del servizio altrove. Ogni ricetta che avvia o riavvia un servizio deve prima definirlo, perciò l'approccio più semplice è indicare in una ricetta separata la definizione del servizio e includerla in altre ricette in base alle esigenze.

> **Note**
>
> Per semplicità, la ricetta di default per questo esempio utilizza una risorsa `service` per avviare il servizio dopo aver eseguito la definizione del servizio. Un'implementazione di produzione in genere avvia o riavvia un servizio utilizzando `notifies`, come illustrato più avanti.

Per eseguire la ricetta

1. Creare un file `default.rb` che contiene l'esempio di ricetta predefinita e salvarlo in `recipes`.
2. Creare un file `service.rb` che contiene l'esempio di definizione del servizio e salvarlo in `recipes`.
3. Eseguire `kitchen converge`, quindi effettuare l'accesso all'istanza ed eseguire il comando seguente per verificare che il servizio sia in esecuzione.

```
sudo service tomcat7 status
```

> **Note**
>
> Se `service.rb` era in esecuzione separatamente da `default.rb`, è necessario modificare `.kitchen.yml` per aggiungere `tomcat::service` all'elenco di esecuzione. Tuttavia, quando si include una ricetta, il relativo codice viene incorporato nella ricetta padre prima che la ricetta viene eseguita. `service.rb` è quindi fondamentalmente una parte di `default.rb` e non richiede un elenco di esecuzione separato.

Utilizzo di notifies per avviare o riavviare un servizio

Di solito, le implementazioni di produzione non utilizzano `service` per avviare o riavviare un servizio. Al contrario, aggiungono `notifies` a una qualsiasi delle diverse risorse. Ad esempio, se vuoi riavviare il servizio dopo la modifica di un file di configurazione, includi `notifies` nella risorsa `template` associata. L'utilizzo di `notifies` presenta i seguenti vantaggi rispetto all'utilizzo di una risorsa `service` per il riavvio esplicito del servizio.

- L'elemento `notifies` riavvia il servizio solo se il file di configurazione associato è stato modificato, eliminando ogni rischio di riavvii superflui del servizio.
- Chef riavvia il servizio al massimo una volta al termine di ogni sessione, indipendentemente dal numero di `notifies` contenuti nell'esecuzione.

  Ad esempio, un'esecuzione Chef può includere più risorse di modelli, ciascuna delle quali modifica un file di configurazione differente e richiede il riavvio del servizio se il file è stato modificato. Tuttavia, in genere è preferibile riavviare il servizio solo una volta, alla fine dell'esecuzione Chef. In caso contrario, può succedere che si tenti di riavviare un servizio non ancora completamente operativo da un riavvio precedente, il che può causare errori.

In questo esempio viene modificato `tomcat::default` per includere una risorsa `template` che utilizza `notifies` per riavviare il servizio. Un esempio realistico potrebbe utilizzare una risorsa template che crea una versione personalizzata di uno dei file di configurazione Tomcat, che sono però alquanto lunghi e complessi. Per semplicità, l'esempio usa soltanto la risorsa template da Creazione di un file da un modello. Non ha niente a che fare con Tomcat, ma consente di illustrare in

modo semplice come utilizzare `notifies`. Per un esempio che illustri come utilizzare i modelli per creare file di configurazione Tomcat, consulta Ricette di impostazione.

Per configurare il libro di ricette

1. Aggiungere la sottodirectory `templates` a `tomcat` e la sottodirectory `default` a `templates`.
2. Copiare il modello `example_data.json.erb` dal libro di ricette `createfile` nella directory `templates/default`.
3. Aggiungere la sottodirectory `attributes` a `tomcat`.
4. Copiare il file attributo `default.rb` dal libro di ricette `createfile` nella directory `attributes`.

La seguente ricetta utilizza `notifies` per riavviare il servizio Tomcat.

```
execute "install_updates" do
  command "apt-get update"
end

package "tomcat7" do
    action :install
end

include_recipe 'tomcat::service'

service 'tomcat' do
  action :enable
end

directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end

template "/srv/www/shared/example_data.json" do
  source "example_data.json.erb"
  mode 0644
  variables(
```

```
    :a_boolean_var => true,
    :a_string_var => "some string"
  )
  only_if {node['createfile']['install_file']}
  notifies :restart, resources(:service => 'tomcat')
end
```

L'esempio unisce la ricetta di Creazione di un file da un modello e la ricetta della sezione precedente, con due modifiche rilevanti:

- La risorsa `service` è ancora presente, ma con uno scopo leggermente diverso.

  L'operazione `:enable` abilita il servizio Tomcat all'avvio.
- La risorsa template ora include `notifies`, che riavvia il servizio Tomcat se `example_data.json` è stato modificato.

  Questo garantisce che il servizio venga avviato quando alla prima installazione di Tomcat e riavviato dopo ogni modifica alla configurazione.

Per eseguire la ricetta

1. Eseguire `kitchen destroy` per iniziare con un'istanza pulita.
2. Sostituire il codice in `default.rb` con l'esempio precedente.
3. Eseguire `kitchen converge`, quindi accedere all'istanza e verificare che il servizio sia in esecuzione.

> **Note**
>
> Se si desidera riavviare un servizio ma la ricetta non include una risorsa, ad esempio `template` che supporta `notifies`, è possibile utilizzare una risorsa fittizia `execute`. Ad esempio
>
> ```
> execute 'trigger tomcat service restart' do
>   command 'bin/true'
>   notifies :restart, resources(:service => 'tomcat')
> end
> ```

La risorsa `execute` deve avere un attributo `command`, anche se viene utilizzata solo come modo per eseguire `notifies`. Questo esempio aggira quel requisito eseguendo `/bin/true`, un comando shell che restituisce semplicemente un codice di riuscita.

Esempio 9: utilizzo di Amazon EC2 Instances

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Fino a questo punto, hai eseguito istanze localmente in. VirtualBox Sebbene sia semplice e veloce, alla fine vorrai testare le tue ricette su un' EC2 istanza Amazon. In particolare, se desideri eseguire ricette su Amazon Linux, è disponibile solo su Amazon EC2. Puoi utilizzare un sistema simile come CentOS per l'implementazione e i test preliminari, ma l'unico modo per testare completamente le tue ricette su Amazon Linux è con un'istanza Amazon EC2 .

Questo argomento mostra come eseguire ricette su un' EC2 istanza Amazon. Utilizzerai Test Kitchen e Vagrant come nelle sezioni precedenti, con due differenze:

- Il driver è `kitchen-ec2` invece di Vagrant.
- Il `.kitchen.yml` file del ricettario deve essere configurato con le informazioni necessarie per avviare l' EC2 istanza Amazon.

Note

In alternativa, è possibile utilizzare il plug-in Vagrant `vagrant-aws`. Par maggiori informazioni, consulta la pagina sul provider AWS Vagrant.

Avrai bisogno delle credenziali AWS per creare un' EC2 istanza Amazon. Se non hai già un account AWS, puoi crearne uno procedendo come segue.

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Parte della procedura di registrazione prevede la ricezione di una telefonata o di un messaggio di testo e l'immissione di un codice di verifica sulla tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Dovresti creare un utente IAM con le autorizzazioni per accedere ad Amazon EC2 e salvare l'accesso e le chiavi segrete dell'utente in una posizione sicura sulla tua workstation. Test Kitchen utilizzerà le credenziali per creare l'istanza. Il modo migliore per fornire le credenziali a Test Kitchen è assegnare le chiavi alle seguenti variabili di ambiente nella workstation.

> **Warning**
>
> Gli utenti IAM dispongono di credenziali a lungo termine, il che rappresenta un rischio per la sicurezza. Per ridurre questo rischio, si consiglia di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari.

- AWS_ACCESS_KEY — la chiave di accesso dell'utente, che avrà un aspetto simile. AKIAIOSFODNN7EXAMPLE
- AWS_SECRET_KEY — la chiave segreta del tuo utente, che avrà un aspetto similewJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY.

Grazie a questo approccio, si riducono le possibilità di compromettere accidentalmente l'account aziendale, ad esempio caricando un progetto contenente le credenziali in un repository pubblico.

Per configurare il libro di ricette

1. Per usare il driver `kitchen-ec2`, è necessario disporre del pacchetto `ruby-dev` installato nel sistema. L'esempio seguente mostra come usare il comando `aptitude` per installare il pacchetto su un sistema Ubuntu.

   ```
   sudo aptitude install ruby1.9.1-dev
   ```

2. Il driver `kitchen-ec2` è un pacchetto gem, che è possibile installare come segue:

   ```
   gem install kitchen-ec2
   ```

   A seconda della workstation, questo comando può richiedere `sudo` oppure è possibile anche utilizzare un gestore di ambiente Ruby come RVM. Questa procedura è stata testata con versione 0.8.0 del driver `kitchen-ec2`, ma sono disponibili versioni più recenti. Per installare una versione specifica, eseguire `gem install kitchen-ec2 -v <version number>`.

3. Devi specificare una coppia di chiavi Amazon EC2 SSH che Test Kitchen può utilizzare per connettersi all'istanza. Se non disponi di una coppia di EC2 chiavi Amazon, consulta Amazon EC2 Key Pairs per informazioni su come crearne una. La coppia di chiavi deve appartenere alla stessa regione AWS dell'istanza. L'esempio utilizza Stati Uniti occidentali (California settentrionale).

Dopo aver selezionato una coppia di chiavi, creare una sottodirectory di `opsworks_cookbooks` denominata `ec2_keys` e copiare il file della chiave privata della coppia di chiavi (`.pem`) in tale sottodirectory. La chiave privata è stata inserita in `ec2_keys` solo per comodità per semplificare il codice, ma può trovarsi ovunque sul tuo sistema.

4. Creare una sottodirectory di `opsworks_cookbooks` denominata `createdir-ec2` e accedervi.
5. Aggiungere un file `metadata.rb` a `createdir-ec2` con il contenuto seguente.

```
name "createdir-ec2"
version "0.1.0"
```

6. Inizializzare Test Kitchen, come descritto in Esempio 1: installazione di pacchetti. La sezione seguente descrive come eseguire la configurazione `.kitchen.yml`, che è molto più complicata per EC2 le istanze Amazon.
7. Aggiungere la sottodirectory `recipes` a `createdir-ec2`.

Configurazione di .kitchen.yml per Amazon EC2

La configurazione viene `.kitchen.yml` effettuata con le informazioni necessarie al `kitchen-ec2` driver per avviare un' EC2 istanza Amazon configurata in modo appropriato. Di seguito è riportato un esempio di `.kitchen.yml` file per un'istanza Amazon Linux nella regione Stati Uniti occidentali (California settentrionale).

```
driver:
  name: ec2
  aws_ssh_key_id: US-East1
  region: us-west-1
  availability_zone: us-west-1c
  require_chef_omnibus: true
  security_group_ids: sg........
  subnet_id: subnet-.........
  associate_public_ip: true
  interface: dns

provisioner:
  name: chef_solo

platforms:
  -name: amazon
```

```
  driver:
    image_id: ami-xxxxxxxx
  transport:
    username: ec2-user
    ssh_key: ../ec2_keys/US-East1.pem

suites:
  - name: default
    run_list:
      - recipe[createdir-ec2::default]
    attributes:
```

È possibile utilizzare le impostazioni di default per le sezioni `provisioner` e `suites`, ma vanno modificate le impostazioni predefinite `driver` e `platforms`. Questo esempio utilizza un elenco minimo di impostazioni e accetta i valori di default per il resto. Per un elenco completo delle `kitchen-ec2` impostazioni, consulta Kitchen: :Ec2: A Test Kitchen Driver per Amazon. EC2

L'esempio seguente imposta i seguenti attributi `driver`. presupponendo che le chiavi segrete e di accesso dell'utente siano state assegnate alle variabili di ambiente standard, come descritto in precedenza. Il driver utilizza tali chiavi per impostazione predefinita. In alternativa, è necessario specificare le chiavi aggiungendo `aws_access_key_id` e `aws_secret_access_key` agli attributi `driver`, impostati sui valori chiave appropriati.

nome

(Obbligatorio) Questo attributo deve essere impostato su `ec2`.

aws_ssh_key_id

(Obbligatorio) Il nome della coppia di chiavi Amazon EC2 SSH, denominato `US-East1` in questo esempio.

transport.ssh_key

(Obbligatorio) Il file della chiave privata (`.pem`) per la chiave specificata per `aws_ssh_key_id`. Per questo esempio, il file è denominato `US-East1.pem` e si trova nella directory `../opsworks/ec2_keys`.

Regione

(Obbligatorio) La regione AWS dell'istanza. L'esempio utilizza US West (California settentrionale), che è rappresentato da`us-west-1`).

availability_zone

(Facoltativo) La zona di disponibilità dell'istanza. Se ometti questa impostazione, Test Kitchen utilizza una zona di disponibilità predefinita per la regione specificata, che corrisponde `us-west-1b` agli Stati Uniti occidentali (California settentrionale). Tuttavia, la zona di default potrebbe non essere disponibile per un determinato account. In tal caso, è necessario specificare una zona di disponibilità. In questa circostanza, l'account utilizzato per preparare gli esempi non supporta `us-west-1b`, per questo l'esempio specifica in modo esplicito `us-west-1c`.

require_chef_omnibus

Se è impostata su `true`, questa impostazione garantisce che il programma di installazione omnibus venga utilizzato per installare `chef-client` su tutte le istanze della piattaforma.

security_group_ids

(Facoltativo) Un elenco di gruppi di sicurezza IDs da applicare all'istanza. Questa impostazione applica il gruppo di sicurezza `default` all'istanza. Assicurati che le regole in ingresso del gruppo di sicurezza consentano connessioni SSH, altrimenti Test Kitchen non sarà in grado di comunicare con l'istanza. Se utilizzi il gruppo di sicurezza `default`, potrebbe essere necessario modificarlo di conseguenza. Per ulteriori informazioni, consulta Amazon EC2 Security Groups.

subnet_id

L'ID della sottorete di destinazione per l'istanza, se applicabile.

associate_public_ip

Puoi fare in modo che Amazon EC2 associ un indirizzo IP pubblico all'istanza se desideri poter accedere all'istanza da Internet.

interface

Il tipo di configurazione del nome host utilizzato per accedere all'istanza. I valori validi sono `dns`, `public`, `private` o `private_dns`. Se non specifichi un valore per questo attributo, `kitchen-ec2` imposta la configurazione del nome host nell'ordine seguente. Se ometti questo attributo, il tipo di configurazione non viene impostato.

1. Nome DNS
2. Indirizzo IP pubblico
3. Indirizzo IP privato
4. Nome DNS privato

> **Important**
>
> Invece di utilizzare le credenziali del tuo account per le chiavi di accesso e segrete, dovresti creare un utente e fornire tali credenziali a Test Kitchen. Per ulteriori informazioni, consulta Best practice per la gestione delle chiavi di accesso AWS.
> Fai attenzione a non inserirlo `.kitchen.yml` in un luogo accessibile al pubblico, ad esempio caricandolo su un repository pubblico GitHub o su Bitbucket. Questo espone le tue credenziali e può compromettere la sicurezza del tuo account.

Il driver `kitchen-ec2` fornisce supporto di default per le seguenti piattaforme:

- ubuntu-10.04
- ubuntu-12.04
- ubuntu-12.10
- ubuntu-13.04
- ubuntu-13.10
- ubuntu-14.04
- centos-6.4
- debian-7.1.0
- windows-2012r2
- windows-2008r2

Se vuoi utilizzare una o più di queste piattaforme, aggiungi i nomi appropriati a `platforms`. Il driver `kitchen-ec2` seleziona automaticamente un'AMI adeguata e genera un nome utente SSH. Puoi usare altre piattaforme, questo esempio utilizza Amazon Linux, ma devi specificare esplicitamente i seguenti attributi. `platforms`

nome

Il nome della piattaforma. Questo esempio usa Amazon Linux, perciò `name` è impostato su `amazon`.

driver

Gli attributi `driver`, che includono quanto segue:

- `image_id`— L'AMI della piattaforma, che deve appartenere alla regione specificata. L'esempio utilizza `ami-ed8e9284` un'AMI Amazon Linux della regione Stati Uniti occidentali (California settentrionale).
- `transport.username`— Il nome utente SSH che Test Kitchen utilizzerà per comunicare con l'istanza.

  Usa `ec2-user` per Amazon Linux. Altri AMIs potrebbero avere nomi utente diversi.

Sostituisci il codice in `.kitchen.yml` con l'esempio, quindi assegna valori appropriati agli attributi specifici dell'account, ad esempio `aws_access_key_id`.

Esecuzione della ricetta

In questo esempio viene utilizzata la ricetta contenuta in Iterazione.

Per eseguire la ricetta

1. Creare un file denominato `default.rb` con il seguente codice e salvarlo nella cartella `recipes` del libro di ricette.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

2. Eseguire `kitchen converge` per eseguire la ricetta. Tieni presente che il completamento di questo comando richiederà più tempo rispetto agli esempi precedenti a causa del tempo necessario per avviare e inizializzare un' EC2 istanza Amazon.
3. Vai alla EC2console Amazon, seleziona la regione Stati Uniti occidentali (California settentrionale) e fai clic su Istanze nel riquadro di navigazione. La nuova istanza sarà visualizzata nell'elenco.
4. Esegui `kitchen login` per accedere all'istanza, proprio come hai fatto per le istanze in esecuzione. VirtualBox Le directory appena create saranno disponibili in `/srv`. È inoltre possibile utilizzare il client SSH preferito per connettersi all'istanza.

Fasi successive

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo capitolo ti ha presentato i concetti di base per l'implementazione dei libri di ricette di Chef, ma c'è molto altro:

- Gli esempi hanno illustrato come utilizzare alcune delle risorse più comuni, ma ce ne sono molte altre.

  Per le risorse trattate, gli esempi hanno utilizzato solo alcune delle operazioni e degli attributi disponibili. Per un riferimento completo, consulta la pagina relativa a risorse e provider.
- Negli esempi sono stati utilizzati gli elementi di base dei libri di ricette, ovvero `recipes`, `attributes`, `files` e `templates`.

  I libri di ricette possono includere anche una serie di altri elementi, ad esempio `libraries`, `definitions` e `specs`. Per ulteriori informazioni, consulta la documentazione di Chef.
- Negli esempi Test Kitchen è stato utilizzato solo come soluzione pratica per avviare istanze, eseguire ricette e accedere alle istanze.

  Test Kitchen è principalmente una piattaforma di testing che è possibile utilizzare per eseguire una serie di test sulle ricette. Se non lo hai già fatto, prosegui con la procedura guidata di Test Kitchen, che ti presenterà le caratteristiche di testing.
- Implementazione di Cookbooks for Stacks AWS OpsWorksfornisce alcuni esempi più avanzati e mostra come implementare i libri di cucina per Stacks. AWS OpsWorks

Implementazione di Cookbooks for Stacks AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

In Nozioni di base sui libri di ricette sono stati introdotti libri di ricette e ricette. Gli esempi in quella sezione erano semplici nella progettazione e funzioneranno su qualsiasi istanza che supporti Chef, comprese le istanze AWS OpsWorks Stacks. Per implementare libri di cucina più sofisticati per AWS OpsWorks Stacks, in genere è necessario sfruttare appieno l'ambiente AWS OpsWorks Stacks, che differisce dallo standard Chef in diversi modi.

Questo argomento descrive le basi dell'implementazione delle ricette per le istanze Stacks. AWS OpsWorks

> **Note**
>
> Se non hai familiarità con le procedure di implementazione dei libri di ricette, ti consigliamo di leggere Nozioni di base sui libri di ricette.

Argomenti

- Esecuzione di una ricetta su un' AWS OpsWorks istanza Stacks Linux
- Esecuzione di una ricetta su un'istanza Windows
- Esecuzione di uno script di Windows PowerShell
- Simulazione degli attributi di configurazione e distribuzione dello stack su Vagrant
- Utilizzo dei valori degli attributi di configurazione e distribuzione dello stack
- Utilizzo di un libro di ricette esterno su un'istanza Linux: Berkshelf
- Utilizzo dell'SDK per Ruby: download di file da Amazon S3
- Installazione di software Windows
- Sostituzione degli attributi integrati

- Sostituzione dei modelli integrati

Esecuzione di una ricetta su un' AWS OpsWorks istanza Stacks Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Test Kitchen e Vagrant forniscono un modo semplice ed efficiente per implementare i libri di cucina, ma per verificare che le ricette di un libro di cucina funzionino correttamente in produzione, devi eseguirle su un'istanza Stacks. AWS OpsWorks In questo argomento viene descritto come installare un libro di ricette personalizzato in un'istanza Linux AWS OpsWorks Stacks ed eseguire una semplice ricetta. L'argomento fornisce inoltre alcuni suggerimenti per correggere in modo efficiente i bug della ricetta.

Per informazioni su come eseguire ricette su istanze Windows, consulta Esecuzione di una ricetta su un'istanza Windows.

Argomenti

- Creazione ed esecuzione della ricetta
- Esecuzione automatica della ricetta
- Risoluzione dei problemi e correzione delle ricette

Creazione ed esecuzione della ricetta

In primo luogo, devi creare uno stack. Di seguito viene brevemente descritto come creare uno stack per questo esempio. Per ulteriori informazioni, consulta Creare un nuovo stack.

Per creare uno stack

1. Aprire la console AWS OpsWorks Stacks e fare clic su Add Stack (Aggiungi stack).
2. Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e fare clic su Add Stack (Aggiungi stack).

- Nome — OpsTest
- Chiave SSH predefinita: una coppia di EC2 chiavi Amazon

Se devi creare una coppia di EC2 chiavi Amazon, consulta Amazon EC2 Key Pairs. La coppia di chiavi deve appartenere alla stessa regione AWS dell'istanza. L'esempio utilizza la regione predefinita degli Stati Uniti occidentali (Oregon).

3. Fare clic su Add a layer (Aggiungi un livello) e aggiungere un livello personalizzato allo stack con le impostazioni seguenti.

   - Nome: OpsTest
   - Nome abbreviato: opstest

   Per gli stack Linux sarà appropriato qualunque tipo di livello, tuttavia l'esempio non richiede nessuno dei pacchetti che vengono installati da altri tipi di livello, pertanto un livello personalizzato è l'approccio più semplice.
4. Aggiungere un'istanza 24/7 con impostazioni predefinite al livello e avviarla.

Durante l'avvio dell'istanza, che in genere richiede diversi minuti, puoi creare il ricettario. Questo esempio utilizza una versione leggermente modificata della ricetta da Logica condizionale, che crea una directory dati il cui nome dipende dalla piattaforma.

Per configurare il libro di ricette

1. Creare una directory in `opsworks_cookbooks` denominata `opstest` e accedervi.
2. Creare un file `metadata.rb` con il seguente contenuto e salvarlo in `opstest`.

```
name "opstest"
version "0.1.0"
```

3. Creare una directory `recipes` in `opstest`.
4. Creare un file `default.rb` con la seguente ricetta e salvarlo nella directory `recipes`.

```
Chef::Log.info("******Creating a data directory.******")
```

```
data_dir = value_for_platform(
  "centos" => { "default" => "/srv/www/shared" },
  "ubuntu" => { "default" => "/srv/www/data" },
  "default" => "/srv/www/config"
)

directory data_dir do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

La ricetta registra un messaggio, ma lo fa chiamando `Chef::Log.info`. Non stai usando Test Kitchen per questo esempio, quindi il metodo non è molto utile. `log` `Chef::Log.info`inserisce il messaggio nel registro di Chef, che puoi leggere al termine dell'esecuzione di Chef. AWS OpsWorks Stacks fornisce un modo semplice per visualizzare questi registri, come descritto più avanti.

> **Note**
>
> I log di Chef in genere contengono una notevole quantità di informazioni di routine, relativamente poco interessanti. I caratteri "*" che racchiudono il testo del messaggio ne semplificano l'identificazione.

5. Creare un archivio `.zip` di `opsworks_cookbooks`. Per installare il tuo ricettario su un'istanza di AWS OpsWorks Stacks, devi archiviarlo in un repository e fornire a AWS OpsWorks Stacks le informazioni necessarie per scaricare il ricettario sull'istanza. I libri di ricette possono essere archiviati in uno qualsiasi dei diversi tipi di repository supportati. Questo esempio memorizza un file di archivio contenente i libri di cucina in un bucket Amazon S3. Per ulteriori informazioni sui repository dei libri di ricette, consulta Repository dei libri di ricette.

   > **Note**
   >
   > Per semplicità, questo esempio archivia l'intera directory `opsworks_cookbooks`. Tuttavia, significa che AWS OpsWorks Stacks scaricherà tutti i libri di cucina presenti nell'`opsworks_cookbooks`istanza, anche se ne utilizzerai solo uno. Per installare solo il libro di ricette di esempio, crea un'altra directory padre in cui sposterai

`opstest`. Quindi crea un archivio `.zip` della directory padre e utilizzalo al posto di `opsworks_cookbooks.zip`.
I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

6. Carica l'archivio in un bucket Amazon S3, rendilo pubblico e registra l'URL dell'archivio.

Adesso puoi installare il libro di ricette ed eseguire la ricetta.

Per eseguire la ricetta

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: S3 Archive
   - URL del repository: l'URL dell'archivio del libro di cucina che hai registrato in precedenza

   Utilizzare i valori predefiniti per altre impostazioni e fare clic su Save (Salva) per aggiornare la configurazione dello stack.
2. Eseguire il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), che installa la versione corrente dei libri di ricette personalizzati nelle istanze dello stack. Se è presente una versione precedente dei libri di cucina, verrà sovrascritta da questo comando.
3. Eseguire la ricetta utilizzando il comando dello stack Execute Recipes (Esegui ricette) con Recipes to execute (Ricette da eseguire) impostato su **`opstest::default`**. Questo comando avvia un'esecuzione di Chef, con un elenco di esecuzione costituito da `opstest::default`.

Dopo aver eseguito correttamente la ricetta, puoi verificarla.

Per verificare opstest

1. Il primo passaggio consiste nell'esaminare il log di Chef. Fare clic su show (mostra) nella colonna Log (Registro) dell'istanza opstest1 per visualizzare il registro. Scorrere e visualizzare il messaggio di log verso il basso.

```
...
[2014-07-31T17:01:45+00:00] INFO: Storing updated cookbooks/opsworks_cleanup/
attributes/customize.rb in the cache.
[2014-07-31T17:01:45+00:00] INFO: Storing updated cookbooks/opsworks_cleanup/
metadata.rb in the cache.
[2014-07-31T17:01:46+00:00] INFO: ******Creating a data directory.******
[2014-07-31T17:01:46+00:00] INFO: Processing template[/etc/hosts] action create
 (opsworks_stack_state_sync::hosts line 3)
...
```

2. Utilizzare SSH per accedere all'istanza ed elencare i contenuti di `/srv/www/`.

Se hai seguito tutti i passaggi, vedrai `/srv/www/config` piuttosto che la directory `/srv/www/shared` prevista. Nella sezione seguente vengono fornite alcune linee guida per la risoluzione rapida di tali bug.

Esecuzione automatica della ricetta

Il comando Execute Recipes (Esegui ricette) è un metodo utile per testare ricette personalizzate. Per tale motivo viene utilizzato nella maggior parte degli esempi. Tuttavia, in pratica di solito esegui le ricette in momenti standard del ciclo di vita di un'istanza, ad esempio dopo il termine dell'avvio dell'istanza o quando distribuisci un'app. AWS OpsWorks Stacks semplifica l'esecuzione di ricette sull'istanza supportando una serie di eventi del ciclo di vita per ogni livello: Setup, Configure, Deploy, Undeploy e Shutdown. Puoi fare in modo che AWS OpsWorks Stacks esegua automaticamente una ricetta sulle istanze di un layer assegnando la ricetta all'evento del ciclo di vita appropriato.

In genere, è opportuno creare le directory non appena terminata la procedura di avvio di un'istanza, che corrisponde all'evento Setup. Di seguito viene mostrato come eseguire la ricetta di esempio al momento dell'installazione, utilizzando lo stesso stack creato in precedenza nell'esempio. Puoi utilizzare la stessa procedura per gli altri eventi.

Per eseguire automaticamente una ricetta al momento della configurazione

1. Scegli Livelli nel pannello di navigazione, quindi scegli l'icona a forma di matita accanto al link Ricette del OpsTest livello.
2. Aggiungere **`opstest::default`** alle ricette Setup (Installazione) del livello, fare clic su + per aggiungerlo al livello e scegliere Save (Salva) per salvare la configurazione.
3. Scegliere Instances (Istanze), aggiungere un'altra istanza a un livello e avviarla.

L'istanza deve essere denominata `opstest2`. Al termine dell'avvio, AWS OpsWorks Stacks verrà eseguito. `opstest::default`

4. Una volta che l'istanza `opstest2` è online, verificare che `/srv/www/shared` sia presente.

> **Note**
>
> Se hai assegnato ricette agli eventi Setup, Configure o Deploy, puoi eseguirle anche manualmente utilizzando un comando dello stack (Setup e Configure) o un comando di distribuzione (Deploy) per attivare l'evento. Se hai più ricette assegnate a un evento, questi comandi le eseguono tutte.

Risoluzione dei problemi e correzione delle ricette

Se non ottieni i risultati previsti o le tue ricette non vengono eseguite correttamente, il processo di risoluzione dei problemi inizia generalmente con l'esame del log di Chef. Contiene una descrizione dettagliata dell'esecuzione e include eventuali messaggi di log inline delle tue ricette. I log sono particolarmente utili se la tua ricetta semplicemente non riesce. In questo caso, i log di Chef registrano l'errore, inclusa una traccia di stack.

Se la ricetta ha esito positivo, come nell'esempio, il log di Chef spesso non si rivela molto utile. In questo caso, puoi individuare il problema solo esaminando più attentamente la ricetta, in particolare le prime righe:

```
Chef::Log.info("******Creating a data directory.******")

data_dir = value_for_platform(
  "centos" => { "default" => "/srv/www/shared" },
  "ubuntu" => { "default" => "/srv/www/data" },
  "default" => "/srv/www/config"
)
...
```

CentOS è un'alternativa ragionevole per Amazon Linux quando testi ricette su Vagrant, tuttavia attualmente le ricette sono in esecuzione su un'istanza Amazon Linux effettiva. Il valore della piattaforma per Amazon Linux è `amazon`, che non è incluso nella chiamata `value_for_platform`,

pertanto la ricetta crea `/srv/www/config` per impostazione predefinita. Per ulteriori informazioni sulla risoluzione dei problemi, consulta Guida al debugging e alla risoluzione dei problemi.

Ora che hai identificato il problema, devi aggiornare la ricetta e verificare la correzione. Puoi tornare ai file sorgente originali, aggiornare`default.rb`, caricare un nuovo archivio su Amazon S3 e così via. Tuttavia, tale processo può risultare lungo e noioso. Di seguito viene mostrato un approccio molto più veloce, particolarmente utile per i bug delle ricette semplici, come quella nell'esempio, che consiste nel modificare la ricetta sull'istanza.

Per modificare una ricetta su un'istanza

1. Utilizzare SSH per accedere all'istanza ed eseguire `sudo su` per innalzare di livello i privilegi. Per accedere alle directory dei libri di ricette, sono necessari privilegi root.
2. AWS OpsWorks Stacks memorizza il tuo libro di cucina in`/opt/aws/opsworks/current/site-cookbooks`, quindi vai a. `/opt/aws/opsworks/current/site-cookbooks/opstest/recipes`

   > **Note**
   >
   > AWS OpsWorks Stacks memorizza anche una copia dei tuoi libri di cucina in. `/opt/aws/opsworks/current/merged-cookbooks` Non modificare tale libro di ricette. Quando esegui la ricetta, AWS OpsWorks Stacks copia il ricettario da `.../site-cookbooks` a`.../merged-cookbooks`, quindi tutte le modifiche apportate verranno sovrascritte. `.../merged-cookbooks`

3. Utilizzare un editor di testo sull'istanza per modificare `default.rb` e sostituire `centos` con `amazon`. La ricetta dovrebbe essere simile alla seguente.

```
Chef::Log.info("******Creating a data directory.******")

data_dir = value_for_platform(
  "amazon" => { "default" => "/srv/www/shared" },
  "ubuntu" => { "default" => "/srv/www/data" },
  "default" => "/srv/www/config"
)
...
```

Per verificare la correzione, esegui la ricetta utilizzando di nuovo il comando dello stack Execute Recipe (Esegui ricetta). L'istanza deve avere una directory `/srv/www/shared`. Se devi apportare ulteriori modifiche a una ricetta, puoi eseguire Execute Recipe (Esegui ricetta) in qualsiasi momento. Non è necessario arrestare e riavviare l'istanza a ogni esecuzione del comando. Una volta che la ricetta funzionerà correttamente, non dimenticare di aggiornare il codice nel libro di ricette.

> **Note**
>
> Se hai assegnato la tua ricetta a un evento del ciclo di vita in modo che AWS OpsWorks Stacks la esegua automaticamente, puoi sempre utilizzare Execute Recipe per eseguire nuovamente la ricetta. Puoi anche eseguire nuovamente la ricetta tutte le volte che vuoi senza riavviare l'istanza utilizzando la console Stacks per attivare manualmente l'evento appropriato. AWS OpsWorks Tuttavia, questo approccio esegue tutte le ricette dell'evento. Ecco un promemoria:
>
> - Utilizza un comando dello stack per attivare gli eventi Setup o Configure.
> - Utilizza un comando di distribuzione per attivare gli eventi Deploy o Undeploy.

Esecuzione di una ricetta su un'istanza Windows

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo argomento è fondamentalmente una versione abbreviata di Esecuzione di una ricetta su un'istanza Linux, che illustra come eseguire una ricetta su uno stack Windows. Ti consigliamo di consultare prima Esecuzione di una ricetta su un'istanza Linux, perché fornisce una discussione più dettagliata, pertinente in gran parte ai due tipi di sistema operativo.

Per una descrizione di come eseguire ricette su istanze AWS OpsWorks Stacks Linux, consulta. Esecuzione di una ricetta su un'istanza Linux

Argomenti

- Abilitazione dell'accesso RDP
- Creazione ed esecuzione della ricetta
- Esecuzione automatica della ricetta

Abilitazione dell'accesso RDP

Prima di iniziare, se non l'hai già fatto, devi configurare un gruppo di sicurezza con una regola in entrata che consente l'accesso RDP per le tue istanze. Tale gruppo sarà necessario al momento della creazione dello stack.

Quando crei il primo stack in una regione, AWS OpsWorks Stacks crea un set di gruppi di sicurezza. Ne includono uno chiamato qualcosa del genere`AWS-OpsWorks-RDP-Server`, che AWS OpsWorks Stacks collega a tutte le istanze di Windows per consentire l'accesso RDP. Tuttavia, per impostazione predefinita, questo gruppo di sicurezza non presenta regole, perciò devi aggiungere una regola in entrata per consentire l'accesso RDP alle istanze.

Per consentire l'accesso RDP

1. Apri la EC2console Amazon, impostala sulla regione dello stack e scegli Security Groups dal pannello di navigazione.
2. Scegli AWS- OpsWorks -RDP-Server, scegli la scheda Inbound e scegli Modifica.
3. Aggiungere una regola con le impostazioni seguenti:

   - Tipo: RDP
   - Sorgente: gli indirizzi IP di origine consentiti.

     In genere vengono permesse richieste RDP in entrata dall'indirizzo IP o da un intervallo di indirizzi IP specificato (di solito l'intervallo di indirizzi IP dell'azienda).

> **Note**
>
> Come descritto di seguito, devi anche modificare le autorizzazioni utente per consentire l'accesso RDP degli utenti standard.

Per ulteriori informazioni, consulta Accesso con RDP.

Creazione ed esecuzione della ricetta

Di seguito viene brevemente descritto come creare uno stack per questo esempio. Per ulteriori informazioni, consulta Creare un nuovo stack.

Creare uno stack

1. Aprire la console AWS OpsWorks Stacks e scegliere Add Stack (Aggiungi stack). Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e scegliere Add Stack (Aggiungi stack).

   - Nome: WindowsRecipeTest
   - Regione: Stati Uniti occidentali (Oregon)

     Questo esempio funzionerà in qualsiasi regione, ma consigliamo di utilizzare Stati Uniti occidentali (Oregon) per i tutorial.
   - Sistema operativo predefinito: Microsoft Windows Server 2012 R2
2. Scegliere Add a layer (Aggiungi un livello) e aggiungere un livello personalizzato allo stack con le impostazioni seguenti.

   - Nome — RecipeTest
   - Nome abbreviato: recipetest
3. Aggiungi un'istanza 24 ore su 24, 7 giorni su 7 con impostazioni predefinite al RecipeTest layer e avviala.

   AWS OpsWorks Stacks si `AWS-OpsWorks-RDP-Server` assegna automaticamente a questa istanza, il che consente agli utenti autorizzati di accedere all'istanza.
4. Scegliere Permissions (Autorizzazioni), poi Edit (Modifica) e SSH/RDP, quindi scegliere sudo/admin. Gli utenti standard necessitano di questa autorizzazione oltre al gruppo di sicurezza `AWS-OpsWorks-RDP-Server` per accedere all'istanza.

> **Note**
>
> Puoi anche effettuare l'accesso come amministratore, ma è necessario eseguire una procedura diversa. Per ulteriori informazioni, consulta Accesso con RDP.

Durante l'avvio dell'istanza, che in genere richiede diversi minuti, puoi creare il ricettario. La ricetta di questo esempio crea una directory dati ed è generalmente la ricetta di Esempio 3: creazione di directory, modificata per Windows.

> **Note**
>
> Quando si implementano i libri di cucina per le istanze di AWS OpsWorks Stacks Windows, si utilizza una struttura di directory leggermente diversa da quella utilizzata quando si implementano i libri di cucina per le istanze Stacks Linux. AWS OpsWorks Per ulteriori informazioni, consulta Repository dei libri di ricette.

Per configurare il libro di ricette

1. Creare una directory denominata `windowstest` e accedervi.
2. Creare un file `metadata.rb` con il seguente contenuto e salvarlo in `windowstest`.

   ```
   name "windowstest"
   version "0.1.0"
   ```

3. Creare una directory `recipes` in `windowstest`.
4. Creare un file `default.rb` con la seguente ricetta e salvarlo nella directory `recipes`.

   ```
   Chef::Log.info("******Creating a data directory.******")

   directory 'C:\data' do
     rights :full_control, 'instance_name\username'
     inherits false
     action :create
   end
   ```

   Sostituisci *username* con il tuo nome utente.
5. Inserire il libro di ricette in un repository.

   Per installare il tuo libro di cucina su un'istanza di AWS OpsWorks Stacks, devi archiviarlo in un repository e fornire a AWS OpsWorks Stacks le informazioni necessarie per scaricare il libro di cucina sull'istanza. Puoi archiviare libri di ricette di Windows come file di archivio in un bucket

S3 o in un repository Git. In questo esempio viene utilizzato un bucket S3, perciò è necessario creare un archivio .zip della directory `windowstest`. Per ulteriori informazioni sui repository dei libri di ricette, consulta Repository dei libri di ricette.

6. Caricare l'archivio in un bucket S3, rendere pubblico l'archivio e registrare il relativo URL. È possibile usare anche un archivio privato. Tuttavia, per questo esempio, un archivio pubblico è sufficiente e molto più facile da utilizzare.

   I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Adesso puoi installare il libro di ricette ed eseguire la ricetta.

Per eseguire la ricetta

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: S3 Archive
   - URL del repository: l'URL dell'archivio del libro di cucina che hai registrato in precedenza

   Accettare i valori predefiniti per altre impostazioni e scegliere Save (Salva) per aggiornare la configurazione dello stack.
2. Eseguire il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), che installa la versione corrente dei libri di ricette personalizzati nelle istanze dello stack, incluse le istanze online. Se è presente una versione precedente dei libri di cucina, verrà sovrascritta da questo comando.
3. Dopo aver aggiornato i libri di ricette personalizzati, eseguire la ricetta eseguendo il comando dello stack Execute Recipes (Esegui ricette) con l'opzione Recipes to execute (Ricette da eseguire) impostata su **`windowstest::default`**. Questo comando avvia un'esecuzione di Chef, con un elenco di esecuzione costituito dalla tua ricetta.

Dopo aver eseguito correttamente la ricetta, puoi verificarla.

Per verificare windowstest

1. Esaminare il log di Chef. Scegliere show (mostra) nella colonna Log (Registro) dell'istanza opstest1 per visualizzare il registro. Scorrere e visualizzare il messaggio di log verso il basso.

```
...
[2014-07-31T17:01:45+00:00] INFO: Storing updated cookbooks/opsworks_cleanup/
attributes/customize.rb in the cache.
[2014-07-31T17:01:45+00:00] INFO: Storing updated cookbooks/opsworks_cleanup/
metadata.rb in the cache.
[2014-07-31T17:01:46+00:00] INFO: ******Creating a data directory.******
[2014-07-31T17:01:46+00:00] INFO: Processing template[/etc/hosts] action create
 (opsworks_stack_state_sync::hosts line 3)
...
```

2. Scegliere Instances (Istanze), quindi rdp nella colonna Actions (Azioni) dell'istanza e richiedere una password RDP con un periodo di scadenza appropriato. Copiare il nome DNS, il nome utente e la password. È quindi possibile utilizzare tali informazioni con un client RDP, ad esempio il client Windows Remote Desktop Connection, per accedere all'istanza e verificare che `c:\data` esista. Per ulteriori informazioni, consulta Accesso con RDP.

> **Note**
>
> Se la tua ricetta non funziona correttamente, consulta Risoluzione dei problemi e correzione delle ricette per suggerimenti relativi alla risoluzione dei problemi (la maggior parte di essi si applica anche alle istanze Windows). Se vuoi testare la correzione modificando la ricetta sull'istanza, cerca il tuo ricettario nella `C:\chef\cookbooks` directory, dove AWS OpsWorks Stacks installa i libri di cucina personalizzati.

Esecuzione automatica della ricetta

Il comando Execute Recipes (Esegui ricette) è un metodo utile per testare ricette personalizzate. Per tale motivo viene utilizzato nella maggior parte degli esempi. Tuttavia, in pratica di solito esegui le ricette in momenti standard del ciclo di vita di un'istanza, ad esempio dopo il termine dell'avvio dell'istanza o quando distribuisci un'app. AWS OpsWorks Stacks semplifica l'esecuzione di ricette sull'istanza supportando una serie di eventi del ciclo di vita per ogni livello: Setup, Configure, Deploy,

Undeploy e Shutdown. Puoi fare in modo che AWS OpsWorks Stacks esegua automaticamente una ricetta sulle istanze di un layer assegnando la ricetta all'evento del ciclo di vita appropriato.

In genere, è opportuno creare le directory non appena terminata la procedura di avvio di un'istanza, che corrisponde all'evento Setup. Di seguito viene mostrato come eseguire la ricetta di esempio al momento dell'installazione, utilizzando lo stesso stack creato in precedenza nell'esempio. Puoi utilizzare la stessa procedura per gli altri eventi.

Per eseguire automaticamente una ricetta al momento della configurazione

1. Scegli Livelli nel pannello di navigazione, quindi scegli l'icona a forma di matita accanto al link Ricette del RecipeTest livello.
2. Aggiungere **`windowstest::default`** alle ricette Setup (Installazione) del livello, scegliere + per aggiungerlo al livello e scegliere Save (Salva) per salvare la configurazione.
3. Scegliere Instances (Istanze), aggiungere un'altra istanza a un livello e avviarla.

   L'istanza deve essere denominata `recipetest2`. Al termine dell'avvio, AWS OpsWorks Stacks verrà eseguito. `windowstest::default`
4. Una volta che l'istanza `recipetest2` è online, verificare che `c:\data` sia presente.

> **Note**
>
> Se hai assegnato ricette agli eventi Setup, Configure o Deploy, puoi eseguirle anche manualmente utilizzando un comando dello stack (Setup e Configure) o un comando di distribuzione (Deploy) per attivare l'evento. Se hai più ricette assegnate a un evento, questi comandi le eseguono tutte.

Esecuzione di uno script di Windows PowerShell

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi esempi presuppongono che tu abbia già completato l'esempio Esecuzione di una ricetta su un'istanza Windows. In caso contrario, ti consigliamo di iniziare da questo esempio, che descrive in particolare come abilitare l'accesso RDP alle tue istanze.

Un modo per fare in modo che una ricetta esegua attività su un'istanza di Windows, in particolare attività che non dispongono di una risorsa Chef corrispondente, è fare in modo che la ricetta esegua uno script di Windows. PowerShell Questa sezione presenta le nozioni di base descrivendo come utilizzare uno script di Windows per installare una funzionalità di Windows. PowerShell

La `powershell_script`risorsa esegue i PowerShell cmdlet di Windows su un'istanza. L'esempio seguente utilizza un WindowsFeature cmdlet Install- per installare un visualizzatore XPS sull'istanza.

Di seguito viene brevemente descritto come creare uno stack per questo esempio. Per ulteriori informazioni, consulta Creare un nuovo stack.

Creare uno stack

1. Aprire la console AWS OpsWorks Stacks e scegliere Add Stack (Aggiungi stack). Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e fare clic su Add Stack (Aggiungi stack).

   - Nome — PowerShellTest
   - Regione: Stati Uniti occidentali (Oregon)

     Questo esempio funzionerà in qualsiasi regione, ma consigliamo di utilizzare Stati Uniti occidentali (Oregon) per i tutorial.
   - Sistema operativo predefinito: Microsoft Windows Server 2012 R2
2. Scegliere Add a layer (Aggiungi un livello) e aggiungere un livello personalizzato allo stack con le impostazioni seguenti.

   - Nome — PowerShell
   - Nome abbreviato: powershell
3. Aggiungi un'istanza 24 ore su 24, 7 giorni su 7 con le impostazioni predefinite al PowerShell layer e avviala.

4. Scegliere Permissions (Autorizzazioni), poi Edit (Modifica) e selezionare SSH/RDP, quindi scegliere sudo/admin. Per accedere all'istanza come utente standard, è necessario disporre di questa autorizzazione oltre al gruppo di sicurezza `AWS-OpsWorks-RDP-Server`.

Durante l'avvio dell'istanza, che in genere richiede diversi minuti, puoi creare il ricettario. La ricetta di questo esempio crea una directory dati ed è generalmente la ricetta di Esempio 3: creazione di directory, modificata per Windows.

Per configurare il libro di ricette

1. Creare una directory denominata `powershell` e accedervi.
2. Creare un file `metadata.rb` con il seguente contenuto e salvarlo in `windowstest`.

```
name "powershell"
version "0.1.0"
```

3. Creare una directory `recipes` all'interno della directory `powershell`.
4. Creare un file `default.rb` con la seguente ricetta e salvarlo nella directory `recipes`.

```
Chef::Log.info("******Installing XPS.******")

powershell_script "Install XPS Viewer" do
  code <<-EOH
    Install-WindowsFeature XPS-Viewer
  EOH
  guard_interpreter :powershell_script
  not_if "(Get-WindowsFeature -Name XPS-Viewer).installed"
end
```

- La risorsa `powershell_script` esegue un cmdlet per installare il visualizzatore XPS.

  In questo esempio viene eseguito un solo cmdlet, ma il blocco `code` può contenere qualsiasi numero di righe di comando.
- L'`guard_interpreter`attributo indica a Chef di utilizzare la versione a 64 bit di Windows. PowerShell
- L'attributo di protezione `not_if` fa sì che Chef non installi la funzione se è già stata installata.

5. Creare un archivio `.zip` della directory `powershell`.
6. Carica l'archivio in un bucket Amazon S3, rendilo pubblico e registra l'URL dell'archivio. È possibile usare anche un archivio privato, tuttavia, per questo esempio, un archivio pubblico è sufficiente e molto più facile da utilizzare.

   I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Adesso puoi installare il libro di ricette ed eseguire la ricetta.

Per eseguire la ricetta

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: S3 Archive
   - URL del repository: l'URL dell'archivio del libro di cucina che hai registrato in precedenza

   Accettare i valori predefiniti per altre impostazioni e scegliere Save (Salva) per aggiornare la configurazione dello stack.
2. Eseguire il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati) per installare la versione corrente dei libri di ricette personalizzati nell'istanza.
3. Al termine dell'operazione Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), eseguire la ricetta tramite il comando dello stack Execute Recipes (Esegui ricette) con l'opzione Recipes to execute (Ricette da eseguire) impostata su **`powershell::default`**.

> **Note**
>
> Questo esempio utilizza Execute Recipes per comodità, ma in genere AWS OpsWorks Stacks esegue le ricette automaticamente assegnandole all'evento del ciclo di vita appropriato. Puoi eseguire tali ricette attivando manualmente l'evento e Puoi utilizzare un comando dello stack per attivare gli eventi Setup e Configure e un comando di distribuzione per attivare gli eventi Deploy e Undeploy.

Dopo aver eseguito correttamente la ricetta, puoi verificarla.

Per verificare la ricetta powershell

1. Esaminare il log di Chef. Fare clic su show (mostra) nella colonna Log (Registro) dell'istanza powershell1 per visualizzare il registro. Scorrere e visualizzare il messaggio di log verso il basso.

```
...
[2015-04-27T18:12:09+00:00] INFO: Storing updated cookbooks/powershell/metadata.rb
 in the cache.
[2015-04-27T18:12:09+00:00] INFO: ******Installing XPS.******
[2015-04-27T18:12:09+00:00] INFO: Processing powershell_script[Install XPS Viewer]
 action run (powershell::default line 3)
[2015-04-27T18:12:09+00:00] INFO: Processing powershell_script[Guard resource]
 action run (dynamically defined)
[2015-04-27T18:12:42+00:00] INFO: powershell_script[Install XPS Viewer] ran
 successfully
...
```

2. Utilizzare RDP per accedere all'istanza e aprire il menu Start (Avvio). Il visualizzatore XPS deve essere elencato con Windows Accessories (Accessori Windows).

Simulazione degli attributi di configurazione e distribuzione dello stack su Vagrant

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo argomento si applica solo alle istanze Linux. Test Kitchen non supporta ancora Windows, quindi eseguirai tutti gli esempi di Windows su AWS OpsWorks istanze Stacks.

AWS OpsWorks Stacks aggiunge gli attributi di configurazione e distribuzione dello stack all'oggetto nodo per ogni istanza dello stack per ogni evento del ciclo di vita. Questi attributi forniscono una snapshot della configurazione dello stack, inclusa la configurazione di ogni livello e le relative istanze online, la configurazione di ogni app distribuita e così via. Poiché questi attributi si trovano nell'oggetto nodo, è possibile accedervi tramite qualsiasi ricetta; la maggior parte delle ricette per le istanze AWS OpsWorks Stacks utilizza uno o più di questi attributi.

Un'istanza in esecuzione in una casella Vagrant non è gestita da AWS OpsWorks Stacks, quindi il suo oggetto nodo non include alcun attributo di configurazione e distribuzione dello stack per impostazione predefinita. Tuttavia, puoi aggiungere una serie di attributi idonei all'ambiente Test Kitchen. Test Kitchen aggiunge quindi gli attributi all'oggetto nodo dell'istanza e le tue ricette possono accedere agli attributi proprio come farebbero su un'istanza Stacks. AWS OpsWorks

In questo argomento viene illustrato come ottenere una copia degli attributi di configurazione e distribuzione di uno stack idoneo, come installare gli attributi in un'istanza e come accedervi.

> **Note**
>
> Se utilizzi Test Kitchen per eseguire test sulle tue ricette, fauxhai fornisce un metodo alternativo per simulare il JSON di configurazione e distribuzione dello stack.

Per configurare il libro di ricette

1. Creare una sottodirectory di `opsworks_cookbooks` denominata `printjson` e accedervi.
2. Inizializzare e configurare Test Kitchen, come descritto in Esempio 1: installazione di pacchetti.
3. Aggiungere due sottodirectory a `printjson`: `recipes` ed `environments`.

Puoi simulare gli attributi di configurazione e distribuzione dello stack aggiungendo un file di attributi al libro di ricette con le definizioni appropriate, tuttavia un approccio migliore consiste nell'utilizzare l'ambiente Test Kitchen. Sono disponibili due approcci di base:

- Aggiungere definizioni degli attributi a `.kitchen.yml`.

  Questo approccio è particolarmente utile se disponi di alcuni attributi. Per ulteriori informazioni, vedi kitchen.yml.
- Definire gli attributi in un file di ambiente e fare riferimento al file in `.kitchen.yml`.

Questo approccio è generalmente preferibile per gli attributi di configurazione e distribuzione dello stack poiché il file di ambiente è già in formato JSON. Puoi ottenere una copia degli attributi in formato JSON da un'istanza AWS OpsWorks Stacks adatta e incollarla. Tutti gli esempi utilizzano un file di ambiente.

Il modo più semplice per creare attributi di configurazione e distribuzione di uno stack per il tuo libro di ricette è realizzare uno stack configurato correttamente e copiare gli attributi risultanti da un'istanza come JSON. Per mantenere il tuo file di ambiente Test Kitchen gestibile, puoi modificare tale JSON in modo da avere solo gli attributi necessari alle tue ricette. Gli esempi in questo capitolo sono basati sullo stack di Nozioni di base sugli stack Linux Chef 11, ovvero lo stack di un server di applicazioni PHP semplice con sistema di bilanciamento del carico, server di applicazioni PHP e server di database MySQL.

Per creare un JSON di configurazione e distribuzione dello stack

1. Crea MyStack come descritto inNozioni di base sugli stack Linux Chef 11, inclusa la distribuzione di Simple. PHPApp Se preferisci, puoi omettere la seconda istanza di PHP App Server richiesta inFase 4: Scalare Out MyStack; gli esempi non utilizzano quegli attributi.
2. Se non è già stato fatto, avviare l'istanza `php-app1`, quindi accedere con SSH.
3. Nella finestra del terminale, eseguire il seguente comando dell'interfaccia a riga di comando dell'agente:

   ```
   sudo opsworks-agent-cli get_json
   ```

   Questo comando visualizza i più recenti attributi di configurazione e distribuzione dello stack più recente dell'istanza nella finestra del terminale in formato JSON.
4. Copiare il JSON in un file `.json` e salvarlo in una posizione comoda sulla workstation. I dettagli dipendono dal client SSH. Ad esempio, se si sta utilizzando PuTTY su Windows, è possibile eseguire il comando `Copy All to Clipboard`, che copia tutto il testo nella finestra del terminale negli Appunti di Windows. È quindi possibile incollare i contenuti in un file `.json` e modificare il file per rimuovere testo estraneo.
5. Modifica MyStack JSON in base alle esigenze. Gli attributi di configurazione e distribuzione dello stack sono numerosi e i libri di ricette utilizzano generalmente solo una piccola parte di essi. Per mantenere gestibile il file di ambiente, è possibile modificare il JSON in modo che conservi la struttura originale ma contenga solo gli attributi che il libro di ricette effettivamente utilizza.

Questo esempio utilizza una versione fortemente modificata di MyStack JSON che include solo due `['opsworks']['stack']` attributi e. `['id] ['name']` Crea una versione modificata di MyStack JSON simile alla seguente:

```
{
  "opsworks": {
    "stack": {
      "name": "MyStack",
      "id": "42dfd151-6766-4f1c-9940-ba79e5220b58",
    },
  },
}
```

Per ottenere questo JSON nell'oggetto nodo dell'istanza, è necessario aggiungerlo a un ambiente Test Kitchen.

Per aggiungere gli attributi di configurazione e distribuzione dello stack all'ambiente Test Kitchen

1. Creare un file di ambiente denominato `test.json` con il seguente contenuto e salvarlo nella cartella `environments` del libro di ricette.

```
{
  "default_attributes": {
    "opsworks" : {
      "stack" : {
        "name" : "MyStack",
        "id" : "42dfd151-6766-4f1c-9940-ba79e5220b58"
      }
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

   Il file di ambiente presenta gli elementi seguenti:

   - `default_attributes`— Gli attributi predefiniti in formato JSON.

Questi attributi vengono aggiunti all'oggetto nodo con il tipo di attributo `default`, ovvero il tipo utilizzato da tutti gli attributi JSON di configurazione e distribuzione dello stack. In questo esempio viene utilizzata la versione modificata del JSON di configurazione e distribuzione dello stack illustrata in precedenza.

- `chef_type`— Imposta questo elemento su. `environment`
- `json_class`— Imposta questo elemento su`Chef::Environment`.

2. Modificare `.kitchen.yml` per definire l'ambiente Test Kitchen, come segue.

```
---
driver:
  name: vagrant

provisioner:
  name: chef_solo
  environments_path: ./environments

platforms:
  - name: ubuntu-12.04

suites:
  - name: printjson
    provisioner:
      solo_rb:
        environment: test
    run_list:
      - recipe[printjson::default]
    attributes:
```

È possibile definire l'ambiente aggiungendo i seguenti elementi al `.kitchen.yml` predefinito creato da `kitchen init`.

provisioner

Aggiungere gli elementi seguenti.

- `name`— Imposta questo elemento su`chef_solo`.

Per replicare più fedelmente l'ambiente AWS OpsWorks Stacks, puoi utilizzare la modalità locale del client Chef anziché Chef solo. La modalità locale è un'opzione del client Chef

che utilizza una versione ridotta del server Chef (Chef Zero), eseguita in locale sull'istanza anziché su un server remoto. Questo consente alle ricette di utilizzare funzioni del server Chef, come la ricerca o i contenitori di dati, senza connettersi a un server remoto.

- `environments_path`— La sottodirectory cookbook che contiene il file di ambiente, `./environments` per questo esempio.

suites:provisioner

Aggiungi un elemento `solo_rb` con una serie di elementi `environment` al nome del file di ambiente, eliminando l'estensione .json. Questo esempio imposta `environment` su `test`.

3. Creare un file di ricetta denominato `default.rb` con il seguente contenuto e salvarlo nella directory `recipes` del libro di ricette.

```
log "Stack name: #{node['opsworks']['stack']['name']}"
log "Stack id: #{node['opsworks']['stack']['id']}"
```

Questa ricetta registra semplicemente i due valori di configurazione e distribuzione dello stack aggiunti all'ambiente. Sebbene la ricetta sia eseguita localmente in Virtual Box, fai riferimento a tali attributi utilizzando la stessa sintassi del nodo che faresti se la ricetta fosse in esecuzione su un' AWS OpsWorks istanza Stacks.

4. Esegui `kitchen converge`. Dovrebbe essere visualizzato un output del log simile al seguente.

```
...
Converging 2 resources
Recipe: printjson::default
  * log[Stack name: MyStack] action write[2014-07-01T23:14:09+00:00] INFO:
 Processing log[Stack name: MyStack] action write (printjson::default line 1)

[2014-07-01T23:14:09+00:00] INFO: Stack name: MyStack

  * log[Stack id: 42dfd151-6766-4f1c-9940-ba79e5220b58] action
 write[2014-07-01T23:14:09+00:00] INFO: Processing log[Stack id:
 42dfd151-6766-4f1c-9940-ba79e5220b58] action write (printjson::default line 2)

[2014-07-01T23:14:09+00:00] INFO: Stack id: 42dfd151-6766-4f1c-9940-ba79e5220b58

...
```

Utilizzo dei valori degli attributi di configurazione e distribuzione dello stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le ricette necessitano spesso di informazioni sulla configurazione dello stack o le app distribuite. Ad esempio, potresti avere bisogno di un elenco degli indirizzi IP dello stack per creare un file di configurazione o della directory di distribuzione di un'app per creare una directory di log. Invece di archiviare questi dati su un server centrale, AWS OpsWorks Stacks installa un set di attributi di configurazione e distribuzione dello stack nell'oggetto nodo di ogni istanza per ogni evento del ciclo di vita. Questi attributi rappresentano l'attuale stato dello stack, incluse le app distribuite. Le ricette possono quindi ottenere i dati di cui hanno bisogno dall'oggetto nodo.

> **Note**
>
> Talvolta le applicazioni necessitano di informazioni dall'oggetto nodo, come i valori degli attributi di configurazione e implementazione dello stack. Tuttavia, un'applicazione non è in grado di accedere all'oggetto nodo. Per fornire i dati dell'oggetto nodo a un'applicazione, puoi implementare una ricetta che recuperi i dati richiesti dall'oggetto nodo e li inserisca in un file in un formato pratico. L'applicazione è quindi in grado di leggere i dati dal file. Per ulteriori informazioni e un esempio, consulta Passaggio di dati alle applicazioni.

Le ricette possono ottenere i valori degli attributi di configurazione e implementazione dello stack dall'oggetto nodo nel modo seguente.

- Direttamente, utilizzando il nome completo di un attributo.

  Puoi usare questo approccio con qualsiasi stack Linux, ma non con gli stack Windows.
- Con la ricerca di Chef, che puoi utilizzare per eseguire query sull'oggetto nodo per i valori degli attributi.

Puoi usare questo approccio con gli stack Windows e Chef 11.10, ma non con gli stack Linux.

> **Note**
>
> Con gli stack Linux, puoi utilizzare l'interfaccia a riga di comando dell'agente per ottenere una copia degli attributi di configurazione e distribuzione dello stack dell'istanza in formato JSON. Per ulteriori informazioni, consulta Simulazione degli attributi di configurazione e distribuzione dello stack su Vagrant.

Argomenti

- Ottenimento diretto dei valori degli attributi
- Ottenimento dei valori degli attributi con la ricerca di Chef

Ottenimento diretto dei valori degli attributi

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo approccio funziona solo per gli stack Linux.

Simulazione degli attributi di configurazione e distribuzione dello stack su Vagrant mostra come ottenere i dati di configurazione e distribuzione dello stack utilizzando la sintassi del nodo per fare riferimento direttamente ad attributi particolari. Questo è talvolta l'approccio migliore. Tuttavia, molti attributi vengono definiti in raccolte o elenchi di cui i contenuti e i nomi possono variare da stack a stack e nel corso del tempo per un determinato stack. Ad esempio, l'attributo `deploy` contiene un

elenco di attributi di app denominati con il nome breve dell'app. Questo elenco, che include i nomi degli attributi dell'app, in genere varia da stack a stack e anche da distribuzione a distribuzione.

Spesso è quindi più utile, e talvolta anche necessario, ottenere i dati richiesti mediante l'enumerazione degli attributi in un elenco o una raccolta. Ad esempio, supponiamo che desideri conoscere gli indirizzi IP pubblici delle istanze del tuo stack. Questa informazione è nell'attributo `['opsworks']['layers']`, che è impostato su una tabella hash contenente un elemento per ogni livello dello stack, denominato con il nome breve del livello. Ogni elemento del livello è impostato su una tabella hash contenente gli attributi del livello, uno dei quali è `['instances']`. A sua volta, questo elemento è impostato su un'ulteriore tabella hash contenente un attributo per ognuna delle istanze del livello, denominate con il nome breve dell'istanza. Ogni attributo dell'istanza è impostato su ancora un'altra tabella hash che contiene gli attributi dell'istanza, incluso `['ip']`, che rappresenta l'indirizzo IP pubblico. In caso di problemi di visualizzazione, la procedura seguente include un esempio in formato JSON.

Questo esempio illustra come ottenere i dati dal JSON di configurazione e distribuzione dello stack per i livelli di uno stack.

Per configurare il libro di ricette

1. Creare una directory in `opsworks_cookbooks` denominata `listip` e accedervi.
2. Inizializzare e configurare Test Kitchen, come descritto in Esempio 1: installazione di pacchetti.
3. Aggiungere due directory a `listip`: `recipes` ed `environments`.
4. Crea una versione JSON modificata degli attributi di MyStack configurazione e distribuzione che contenga gli attributi pertinenti. Avrà un aspetto simile al seguente.

```
{
  "opsworks": {
    "layers": {
      "php-app": {
        "name": "PHP App Server",
        "id": "efd36017-ec42-4423-b655-53e4d3710652",
        "instances": {
          "php-app1": {
            "ip": "192.0.2.0"
          }
        }
      },
      "db-master": {
```

```
        "name": "MySQL",
        "id": "2d8e0b9a-0d29-43b7-8476-a9b2591a7251",
        "instances": {
          "db-master1": {
            "ip": "192.0.2.5"
          }
        }
      },
      "lb": {
        "name": "HAProxy",
        "id": "d5c4dda9-2888-4b22-b1ea-6d44c7841193",
        "instances": {
          "lb1": {
            "ip": "192.0.2.10"
          }
        }
      }
    }
  }
}
```

5. Creare un file di ambiente denominato `test.json`, incollare il JSON di esempio in `default_attributes` e salvare il file nella cartella `environments` del libro di ricette. Il file deve avere un aspetto simile al seguente (per brevità, la maggior parte del JSON di esempio è rappresentato da un'ellissi).

```
{
  "default_attributes" : {
    "opsworks": {
      "layers": {
        ...
      }
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

6. Sostituire il testo in `.kitchen.yml` con il seguente.

```
---
```

```
driver:
  name: vagrant

provisioner:
  name: chef_zero
  environments_path: ./environment

platforms:
  - name: ubuntu-12.04

suites:
  - name: listip
    provisioner:
      client_rb:
        environment: test
    run_list:
      - recipe[listip::default]
    attributes:
```

Dopo aver impostato il ricettario, puoi utilizzare la seguente ricetta per registrare il layer. IDs

```
node['opsworks']['layers'].each do |layer, layerdata|
  log "#{layerdata['name']} : #{layerdata['id']}"
end
```

La ricetta enumera i livelli in `['opsworks']['layers']` e registra il nome e l'ID di ogni livello.

Per eseguire la ricetta di registrazione degli ID di livello

1. Creare un file denominato `default.rb` con la ricetta di esempio e salvarlo nella directory `recipes`.
2. Esegui `kitchen converge`.

La parte pertinente dell'output deve essere simile a quanto riportato di seguito.

```
Recipe: listip::default
```

```
  * log[PHP App Server : efd36017-ec42-4423-b655-53e4d3710652] action
 write[2014-07-17T22:56:19+00:00] INFO: Processing log[PHP App Server : efd36017-
ec42-4423-b655-53e4d3710652] action write (listip::default line 4)
[2014-07-17T22:56:19+00:00] INFO: PHP App Server : efd36017-ec42-4423-b655-53e4d3710652



  * log[MySQL : 2d8e0b9a-0d29-43b7-8476-a9b2591a7251] action
 write[2014-07-17T22:56:19+00:00] INFO: Processing log[MySQL : 2d8e0b9a-0d29-43b7-8476-
a9b2591a7251] action write (listip::default line 4)
[2014-07-17T22:56:19+00:00] INFO: MySQL : 2d8e0b9a-0d29-43b7-8476-a9b2591a7251


  * log[HAProxy : d5c4dda9-2888-4b22-b1ea-6d44c7841193] action
 write[2014-07-17T22:56:19+00:00] INFO: Processing log[HAProxy : d5c4dda9-2888-4b22-
b1ea-6d44c7841193] action write (listip::default line 4)
[2014-07-17T22:56:19+00:00] INFO: HAProxy : d5c4dda9-2888-4b22-b1ea-6d44c7841193
```

Per visualizzare l'elenco degli indirizzi IP delle istanze, sarà necessario un ciclo nidificato come il seguente.

```
node['opsworks']['layers'].each do |layer, layerdata|
  log "#{layerdata['name']} : #{layerdata['id']}"
  layerdata['instances'].each do |instance, instancedata|
    log "Public IP: #{instancedata['ip']}"
  end
end
```

Il ciclo interno esegue iterazioni sulle istanze di ciascun livello e registra gli indirizzi IP.

Per eseguire la ricetta di registrazione degli IP dell'istanza

1. Sostituire il codice in `default.rb` con la ricetta di esempio.
2. Eseguire `kitchen converge` per eseguire la ricetta.

La parte pertinente dell'output deve essere simile a quanto riportato di seguito.

```
  * log[PHP App Server : efd36017-ec42-4423-b655-53e4d3710652] action
 write[2014-07-17T23:09:34+00:00] INFO: Processing log[PHP App Server : efd36017-
ec42-4423-b655-53e4d3710652] action write (listip::default line 2)
[2014-07-17T23:09:34+00:00] INFO: PHP App Server : efd36017-ec42-4423-b655-53e4d3710652



  * log[Public IP: 192.0.2.0] action write[2014-07-17T23:09:34+00:00] INFO: Processing
 log[Public IP: 192.0.2.0] action write (listip::default line 4)
[2014-07-17T23:09:34+00:00] INFO: Public IP: 192.0.2.0


  * log[MySQL : 2d8e0b9a-0d29-43b7-8476-a9b2591a7251] action
 write[2014-07-17T23:09:34+00:00] INFO: Processing log[MySQL : 2d8e0b9a-0d29-43b7-8476-
a9b2591a7251] action write (listip::default line 2)
[2014-07-17T23:09:34+00:00] INFO: MySQL : 2d8e0b9a-0d29-43b7-8476-a9b2591a7251


  * log[Public IP: 192.0.2.5] action write[2014-07-17T23:09:34+00:00] INFO: Processing
 log[Public IP: 192.0.2.5] action write (listip::default line 4)
[2014-07-17T23:09:34+00:00] INFO: Public IP: 192.0.2.5


  * log[HAProxy : d5c4dda9-2888-4b22-b1ea-6d44c7841193] action
 write[2014-07-17T23:09:34+00:00] INFO: Processing log[HAProxy : d5c4dda9-2888-4b22-
b1ea-6d44c7841193] action write (listip::default line 2)
[2014-07-17T23:09:34+00:00] INFO: HAProxy : d5c4dda9-2888-4b22-b1ea-6d44c7841193


  * log[Public IP: 192.0.2.10] action write[2014-07-17T23:09:34+00:00] INFO: Processing
 log[Public IP: 192.0.2.10] action write (listip::default line 4)
[2014-07-17T23:09:34+00:00] INFO: Public IP: 192.0.2.10
```

Al termine, eseguire `kitchen destroy`; l'argomento successivo utilizza un nuovo libro di ricette.

> **Note**
>
> Uno dei motivi più comuni per l'enumerazione di una raccolta JSON di configurazione e distribuzione di uno stack è ottenere i dati di una determinata app distribuita, ad esempio la relativa directory di distribuzione. Per vedere un esempio, consulta Ricette di ditribuzione.

Ottenimento dei valori degli attributi con la ricerca di Chef

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo approccio è disponibile per gli stack Windows e gli stack Linux Chef 11.10.

Ottenere i valori degli attributi di configurazione e distribuzione dello stack direttamente dall'oggetto nodo può essere complicato e non è una procedura utilizzabile con gli stack Windows. Un approccio alternativo consiste nell'utilizzare la ricerca di Chef per eseguire query per gli attributi di interesse. Se conosci il server Chef, scoprirai che Chef search funziona in modo leggermente diverso con AWS OpsWorks Stacks. Poiché AWS OpsWorks Stacks utilizza chef-client in modalità locale, Chef search dipende da una versione locale del server Chef chiamata chef-zero, in modo che la ricerca operi sui dati archiviati localmente nell'oggetto nodo dell'istanza anziché su un server remoto.

In pratica, limitare la ricerca ai dati archiviati localmente di solito non ha importanza perché l'oggetto node su un' AWS OpsWorks istanza Stacks include gli attributi di configurazione e distribuzione dello stack. Contengono la maggior parte se non tutti i dati che le ricette ottengono in genere dal server Chef e utilizzano gli stessi nomi, quindi di solito è possibile utilizzare il codice di ricerca scritto per Chef server su AWS OpsWorks istanze Stacks senza modifiche. Per ulteriori informazioni, consulta Utilizzo della ricerca di Chef.

Di seguito viene mostrata la struttura di base di una query di ricerca:

```
result = search(:search_index, "key:pattern")
```

- L'indice di ricerca specifica gli attributi a cui si applica la query e determina il tipo di oggetto restituito.
- La chiave specifica il nome dell'attributo.

- Il modello specifica i valori dell'attributo che intendi recuperare.

  Puoi eseguire query per valori di attributi specifici oppure utilizzare i caratteri jolly per eseguire query per un'ampia gamma di valori.
- Il risultato è un elenco di oggetti che soddisfano la query. Ciascuno di questi è una tabella hash contenente più attributi correlati.

  Ad esempio, se utilizzi l'indice di ricerca `node`, la query restituisce un elenco di oggetti istanza, uno per ogni istanza che soddisfa la query. Ogni oggetto è una tabella hash che contiene un set di attributi che definiscono la configurazione dell'istanza, ad esempio il nome host e l'indirizzo IP.

Ad esempio, la query seguente utilizza l'indice di ricerca `node`, ovvero un indice Chef standard che si applica alle istanze dello stack (o nodi, nella terminologia Chef). Questo consente di cercare le istanze con nome host `myhost`.

```
result = search(:node, "hostname:myhost")
```

La ricerca restituisce un elenco di oggetti istanza il cui nome host è `myhost`. Il sistema operativo della prima istanza, ad esempio, sarà rappresentato da `result[0][:os]`. Se la query restituisce più oggetti, puoi enumerarli per recuperare le informazioni richieste.

I dettagli relativi a come utilizzare la ricerca in una ricetta variano a seconda che tu stia utilizzando uno stack Linux o Windows. I seguenti argomenti forniscono esempi in merito a entrambi i tipi di stack.

Argomenti

- Utilizzo della ricerca su uno stack Linux
- Utilizzo della ricerca su uno stack Windows

Utilizzo della ricerca su uno stack Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo esempio si basa su uno stack Linux con un unico server di applicazioni PHP. Utilizza la ricerca di Chef per ottenere l'indirizzo IP pubblico del server e inserisce l'indirizzo in un file nella directory `/tmp`. Recupera essenzialmente le stesse informazioni dall'oggetto nodo di Ottenimento diretto dei valori degli attributi , ma il codice è molto più semplice e non dipende dai dettagli della struttura degli attributi di configurazione e distribuzione dello stack.

Di seguito viene brevemente descritto come creare lo stack per questo esempio. Per ulteriori informazioni, consulta Creare un nuovo stack.

> **Note**
>
> Se non hai mai eseguito una ricetta personalizzata su un'istanza AWS OpsWorks Stacks, dovresti prima seguire l'esempio. Esecuzione di una ricetta su un'istanza Linux

Creare uno stack

1. Aprire la console AWS OpsWorks Stacks e fare clic su Add Stack (Aggiungi stack).
2. Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e fare clic su Add Stack (Aggiungi stack).

   - Nome: searchJSON
   - Chiave SSH predefinita: una coppia di EC2 chiavi Amazon

   Se devi creare una coppia di EC2 chiavi Amazon, consulta Amazon EC2 Key Pairs. La coppia di chiavi deve appartenere alla stessa regione AWS dell'istanza. L'esempio utilizza la regione Stati Uniti occidentali (Oregon).
3. Fai clic su Aggiungi un livello e aggiungi un livello PHP App Server allo stack con le impostazioni predefinite.
4. Aggiungere un'istanza 24/7 con impostazioni predefinite al livello e avviarla.

Per configurare il libro di ricette

1. Creare una directory in `opsworks_cookbooks` denominata `searchjson` e accedervi.
2. Creare un file `metadata.rb` con il seguente contenuto e salvarlo in `opstest`.

```
name "searchjson"
version "0.1.0"
```

3. Creare una directory `recipes` in `searchjson`.
4. Creare un file `default.rb` con la seguente ricetta e salvarlo nella directory `recipes`.

```
phpserver = search(:node, "layers:php-app").first
Chef::Log.info("**********The public IP address is: '#{phpserver[:ip]}'**********")

file "/tmp/ip_addresses" do
  content "#{phpserver[:ip]}"
  mode 0644
  action :create
end
```

   Gli stack Linux supportano solo l'indice di ricerca `node`. La ricetta utilizza questo indice per ottenere un elenco di istanze nel livello `php-app`. Poiché il livello è noto per avere una sola istanza, la ricetta assegna semplicemente la prima a `phpserver`. Se il livello ha più istanze, puoi enumerarle per recuperare le informazioni richieste. Ogni voce dell'elenco è una tabella hash contenente un set di attributi di istanza. L'attributo `ip` è impostato sull'indirizzo IP pubblico dell'istanza, pertanto puoi rappresentare tale indirizzo nel codice della ricetta successiva nella forma `phpserver[:ip]`.

   Dopo aver aggiunto un messaggio al log di Chef, la ricetta usa una risorsa `file` per creare un file denominato `ip_addresses`. L'attributo `content` è impostato su una rappresentazione di stringa di `phpserver[:ip]`. Quando Chef crea `ip_addresses`, aggiunge la stringa al file.

5. Crea un `.zip` archivio di`opsworks_cookbooks`, carica l'archivio in un bucket Amazon S3, rendi l'archivio pubblico e registra l'URL dell'archivio. Per ulteriori informazioni sui repository dei libri di ricette, consulta Repository dei libri di ricette.

I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Adesso puoi installare il libro di ricette ed eseguire la ricetta.

Per eseguire la ricetta

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: Http Archive
   - URL del repository: l'URL dell'archivio del libro di cucina che hai registrato in precedenza

   Utilizzare i valori predefiniti per altre impostazioni e fare clic su Save (Salva) per aggiornare la configurazione dello stack.
2. Modifica la configurazione personalizzata del livello e assegnala `searchjson::default` all'evento Setup del livello. AWS OpsWorks Stacks eseguirà la ricetta dopo l'avvio dell'istanza o se attivi esplicitamente l'evento Setup.
3. Eseguire il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), che installa la versione corrente del repository dei libri di ricette personalizzati nelle istanze dello stack. Se è presente una versione precedente del repository, verrà sovrascritta da questo comando.
4. Eseguire la ricetta utilizzando il comando dello stack Setup (Installazione), che attiva un evento Setup sull'istanza ed esegue `searchjson::default`. Lasciare aperta la pagina Running command setup (Esecuzione configurazione comando).

Dopo aver eseguito correttamente la ricetta, puoi verificarla.

Per verificare searchjson

1. Il primo passo è esaminare il log di Chef per l'evento Setup più recente. Nella pagina di configurazione dell'esecuzione del comando, fare clic su show (mostra) nella colonna Log dell'istanza php-app1 per visualizzare il log. Scorrere verso il basso per trovare il messaggio di log verso la metà. Sarà simile al seguente.

```
...
[2014-09-05T17:08:41+00:00] WARN: Previous
 bash[logdir_existence_and_restart_apache2]: ...
[2014-09-05T17:08:41+00:00] WARN: Current
 bash[logdir_existence_and_restart_apache2]: ...
[2014-09-05T17:08:41+00:00] INFO: **********The public IP address is:
 '192.0.2.0'**********
[2014-09-05T17:08:41+00:00] INFO: Processing directory[/etc/sysctl.d] action create
 (opsworks_initial_setup::sysctl line 1)
...
```

2. Utilizzare SSH per accedere all'istanza ed elencare i contenuti di `/tmp`, che dovrebbero includere un file denominato `ip_addresses` contenente l'indirizzo IP.

Utilizzo della ricerca su uno stack Windows

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks offre due opzioni per utilizzare la ricerca negli stack di Windows.

- L'indice di ricerca `node`, che può essere utilizzato per eseguire query su un set di attributi standard di Chef.

  Se disponi di ricette esistenti con il codice di ricerca che utilizzano`node`, di solito funzionano sugli AWS OpsWorks stack di Stacks senza modifiche.

- È possibile utilizzare un ulteriore set di indici di ricerca per eseguire query su set di attributi di AWS OpsWorks Stacks specifici e alcuni attributi standard.

  Questi indici vengono illustrati in Utilizzo di indici di ricerca AWS OpsWorks specifici per Stacks su Windows Stacks.

Ti consigliamo di usare node per recuperare informazioni standard, come nomi host o indirizzi IP. Questo approccio manterrà la tue ricette conformi alle pratiche standard di Chef. Usa gli indici di ricerca di AWS OpsWorks Stacks per recuperare informazioni specifiche di Stacks. AWS OpsWorks

Argomenti

- Utilizzo dell'indice di ricerca del nodo sugli stack Windows
- Utilizzo di indici di ricerca AWS OpsWorks specifici per Stacks su Windows Stacks

Utilizzo dell'indice di ricerca del nodo sugli stack Windows

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Note

Questo esempio presuppone che tu abbia già completato l'esempio Esecuzione di una ricetta su un'istanza Windows. In caso contrario, ti consigliamo di iniziare da questo esempio, che descrive in particolare come abilitare l'accesso RDP alle istanze.

Questo esempio si basa su uno stack Windows con un singolo livello personalizzato e un'istanza. Uilizza la ricerca di Chef con l'indice di ricerca `node` per ottenere l'indirizzo IP pubblico del server e inserisce l'indirizzo in un file nella directory `C:\tmp`. Di seguito viene brevemente descritto come creare lo stack per questo esempio. Per ulteriori informazioni, consulta Creare un nuovo stack.

Creare uno stack

1. Aprire la console AWS OpsWorks Stacks e scegliere Add Stack (Aggiungi stack).
2. Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e scegliere Add Stack (Aggiungi stack).
   - Nome — NodeSearch

- Regione: Stati Uniti occidentali (Oregon)

  Questo esempio funzionerà in qualsiasi regione, ma consigliamo di utilizzare Stati Uniti occidentali (Oregon) per i tutorial.

- Sistema operativo predefinito: Microsoft Windows Server 2012 R2

3. Scegliere Add a layer (Aggiungi un livello) e aggiungere un livello personalizzato allo stack con le impostazioni seguenti.

   - Nome — IPTest
   - Nome breve: iptest

4. Aggiungi un'istanza t2.micro 24 ore su 24, 7 giorni su 7 con impostazioni predefinite al IPTest layer e avviala. Verrà denominata iptest1.

   AWS OpsWorks Stacks si `AWS-OpsWorks-RDP-Server` assegna automaticamente a questa istanza, il che consente agli utenti autorizzati di accedere all'istanza.

5. Scegliere Permissions (Autorizzazioni), poi Edit (Modifica) e selezionare SSH/RDP, quindi scegliere sudo/admin. Gli utenti standard necessitano di questa autorizzazione oltre al gruppo di sicurezza `AWS-OpsWorks-RDP-Server` per accedere all'istanza.

   > **Note**
   >
   > Puoi anche effettuare l'accesso come amministratore, ma è necessario eseguire una procedura diversa. Per ulteriori informazioni, consulta Accesso con RDP.

Per configurare il libro di ricette

1. Creare una directory denominata `nodesearch` e accedervi.
2. Creare un file `metadata.rb` con il seguente contenuto e salvarlo in `opstest`.

   ```
   name "nodesearch"
   version "0.1.0"
   ```

3. Creare una directory `recipes` in `nodesearch`.
4. Creare un file `default.rb` con la seguente ricetta e salvarlo nella directory `recipes`.

```
directory 'C:\tmp' do
  rights :full_control, 'Everyone'
  recursive true
  action :create
end

windowsserver = search(:node, "hostname:iptest*").first
Chef::Log.info("**********The public IP address is:
 '#{windowsserver[:ipaddress]}'**********")

file 'C:\tmp\addresses.txt' do
  content "#{windowsserver[:ipaddress]}"
  rights :full_control, 'Everyone'
  action :create
end
```

La ricetta effettua le seguenti operazioni:

1. Utilizza una risorsa di directory per creare una directory `C:\tmp` per il file.

   Per ulteriori informazioni su questa risorsa, consulta Esempio 3: creazione di directory.

2. Utilizza la ricerca di Chef con l'indice di ricerca `node` per ottenere un elenco di nodi (istanze) con un nome host che inizia per `iptest`.

   Se si utilizza il tema predefinito, che crea nomi host aggiungendo numeri interi al nome breve del layer, questa query restituirà ogni istanza del layer. IPTest In questo esempio il livello ha una sola istanza, pertanto la ricetta assegna semplicemente la prima a `windowsserver`. Per più istanze, puoi ottenere un elenco completo, quindi puoi enumerarle.

3. Aggiunge un messaggio con l'indirizzo IP del log di Chef per questa esecuzione.

   L'oggetto `windowsserver` è una tabella hash il cui attributo `ipaddress` è impostato sull'indirizzo IP pubblico dell'istanza, pertanto puoi rappresentare tale indirizzo nel codice della ricetta successiva nella forma `windowsserver[:ipaddress]`. La ricetta inserisce la stringa corrispondente nel messaggio e la aggiunge al log di Chef.

4. Utilizza la risorsa `file` per creare un file con l'indirizzo IP denominato `C:\tmp \addresses.txt`.

L'attributo `content` della risorsa specifica contenuto da aggiungere al file, ovvero, in questo caso, l'indirizzo IP pubblico.

5. Creare un archivio `.zip` di `nodesearch`, Caricare l'archivio in un bucket S3, rendere pubblico l'archivio e registrare il relativo URL.

   I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Adesso puoi installare il libro di ricette ed eseguire la ricetta.

Per installare il libro di ricette ed eseguire la ricetta

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: S3 Archive
   - URL del repository: l'URL dell'archivio del libro di cucina che hai registrato in precedenza

   Accettare i valori predefiniti per altre impostazioni e scegliere Save (Salva) per aggiornare la configurazione dello stack.
2. Eseguire il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), che installa la versione corrente dei libri di ricette personalizzati nelle istanze dello stack, incluse le istanze online. Se è presente una versione precedente dei libri di cucina, verrà sovrascritta da questo comando.
3. Al termine dell'operazione Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), eseguire la ricetta tramite il comando dello stack Execute Recipes (Esegui ricette) con l'opzione Recipes to execute (Ricette da eseguire) impostata su **`nodesearch::default`**. Questo comando avvia un'esecuzione di Chef, con un elenco di esecuzione costituito dalla tua ricetta. Lasciare la pagina execute_recipes aperta.

Dopo aver eseguito correttamente la ricetta, puoi verificarla.

Per verificare nodesearch

1. Esaminare il log di Chef per l'evento execute_recipes più recente. Nella pagina di esecuzione del comando execute_recipes, scegliere show (mostra) nella colonna Log dell'istanza iptest1 per visualizzare il log. Scorrere verso il basso per trovare il messaggio di log, che sarà simile al seguente.

```
...
[2015-05-13T18:55:47+00:00] INFO: Storing updated cookbooks/nodesearch/recipes/
default.rb in the cache.
[2015-05-13T18:55:47+00:00] INFO: Storing updated cookbooks/nodesearch/metadata.rb
 in the cache.
[2015-05-13T18:55:47+00:00] INFO: **********The public IP address is:
 '192.0.0.1'**********
[2015-05-13T18:55:47+00:00] INFO: Processing directory[C:\tmp] action create
 (nodesearch::default line 1)
[2015-05-13T18:55:47+00:00] INFO: Processing file[C:\tmp\addresses.txt] action
 create (nodesearch::default line 10)
...
```

2. Utilizzare RDP per accedere all'istanza ed esaminare i contenuti di `C:\tmp\addresses.txt`.

Utilizzo di indici di ricerca AWS OpsWorks specifici per Stacks su Windows Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo esempio presuppone che tu abbia già completato l'esempio Esecuzione di una ricetta su un'istanza Windows. In caso contrario, ti consigliamo di iniziare da questo esempio, che descrive in particolare come abilitare l'accesso RDP alle istanze.

AWS OpsWorks Stacks fornisce i seguenti indici di ricerca oltre a: `node`

- `aws_opsworks_stack`— La configurazione dello stack.
- `aws_opsworks_layer`— Le configurazioni dei livelli dello stack.
- `aws_opsworks_instance`— Le configurazioni delle istanze dello stack.
- `aws_opsworks_app`— Le configurazioni delle app dello stack.
- `aws_opsworks_user`— Le configurazioni utente dello stack.
- `aws_opsworks_rds_db_instance`— Informazioni di connessione per le istanze RDS registrate.

Questi indici includono alcuni attributi Chef standard, ma sono destinati principalmente al recupero di attributi specifici di Stacks. AWS OpsWorks Ad esempio `aws_opsworks_instance` include un attributo `status` che fornisce lo stato dell'istanza, come `online`.

> **Note**
>
> La procedura consigliata consiste nell'utilizzare `node` quando possibile per mantenere le ricette conformi all'utilizzo standard di Chef. Per vedere un esempio, consulta Utilizzo dell'indice di ricerca del nodo sugli stack Windows.

Questo esempio mostra come utilizzare gli indici Stacks per recuperare il valore di un attributo specifico di AWS OpsWorks Stacks. AWS OpsWorks Si basa su uno stack Windows semplice con un livello personalizzato che include un'istanza. Utilizza la ricerca Chef per ottenere l'ID AWS OpsWorks Stacks dell'istanza e inserisce i risultati nel registro Chef.

Di seguito viene brevemente descritto come creare uno stack per questo esempio. Per ulteriori informazioni, consulta Creare un nuovo stack.

Creare uno stack

1. Aprire la console AWS OpsWorks Stacks e scegliere + Stack. Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e scegliere Add Stack (Aggiungi stack).

   - Nome: IDSearch
   - Regione: Stati Uniti occidentali (Oregon)

     Questo esempio funzionerà in qualsiasi regione, ma consigliamo di utilizzare Stati Uniti occidentali (Oregon) per i tutorial.

- Sistema operativo predefinito: Microsoft Windows Server 2012 R2

2. Scegliere Add a layer (Aggiungi un livello) e aggiungere un livello personalizzato allo stack con le impostazioni seguenti.

   - Nome — IDCheck
   - Nome abbreviato: idcheck

3. Aggiungi un'istanza t2.micro 24 ore su 24, 7 giorni su 7 con impostazioni predefinite al IDCheck livello e avviala. Verrà denominata iptest1.

   AWS OpsWorks Stacks si `AWS-OpsWorks-RDP-Server` assegna automaticamente a questa istanza. Abilitazione dell'accesso RDPspiega come aggiungere una regola in entrata a questo gruppo di sicurezza che consenta agli utenti autorizzati di accedere all'istanza.

4. Scegliere Permissions (Autorizzazioni), poi Edit (Modifica) e SSH/RDP, quindi scegliere sudo/admin. Gli utenti standard necessitano di questa autorizzazione oltre al gruppo di sicurezza `AWS-OpsWorks-RDP-Server` per accedere all'istanza.

> **Note**
>
> Puoi anche effettuare l'accesso come amministratore, ma è necessario eseguire una procedura diversa. Per ulteriori informazioni, consulta Accesso con RDP.

Per configurare il libro di ricette

1. Creare una directory denominata `idcheck` e accedervi.
2. Creare un file `metadata.rb` con il seguente contenuto e salvarlo in `opstest`.

```
name "idcheck"
version "0.1.0"
```

3. Creare una directory `recipes` in `idcheck` e aggiungere un file `default.rb` alla directory che contiene la seguente ricetta.

```
windowsserver = search(:aws_opsworks_instance, "hostname:idcheck*").first
Chef::Log.info("**********The public IP address is:
 '#{windowsserver[:instance_id]}'**********")
```

La ricetta usa la ricerca di Chef con un indice di ricerca `aws_opsworks_instance` per ottenere gli attributi di istanza di ciascuna istanza nello stack con un nome host che inizia per `idcheck`. Se si utilizza il tema predefinito, che crea nomi host aggiungendo numeri interi al nome breve del layer, questa query restituirà ogni istanza del layer. IDCheck In questo esempio il livello ha una sola istanza, pertanto la ricetta assegna semplicemente la prima a `windowsserver`. Per più istanze, puoi ottenere un elenco completo, quindi puoi enumerarle.

La ricetta sfrutta il fatto che esiste una sola istanza nello stack con questo nome host, pertanto il primo risultato è quello corretto. Se il tuo stack dispone di più istanze, la ricerca su altri attributi può restituire più di un risultato. Per un elenco di attributi di istanza, consulta Contenitore di dati dell'istanza (aws_opsworks_instance).

Gli attributi dell'istanza sono fondamentalmente una tabella hash e all'`instance_id`attributo viene assegnato AWS OpsWorks lo Stacks ID dell'istanza, quindi potete fare riferimento all'ID come. `windowsserver[:instance_id]` La ricetta inserisce la stringa corrispondente nel messaggio e la aggiunge al log di Chef.

4. Crea un `.zip` archivio del `ipaddress` libro di cucina, carica l'archivio in un bucket Amazon S3 e registra l'URL dell'archivio. Per ulteriori informazioni sui repository dei libri di ricette, consulta Repository dei libri di ricette.

   I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Adesso puoi installare il libro di ricette ed eseguire la ricetta.

Per installare il libro di ricette ed eseguire la ricetta

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: S3 Archive
   - URL del repository: l'URL dell'archivio del libro di cucina che hai registrato in precedenza

   Accettare i valori predefiniti per altre impostazioni e scegliere Save (Salva) per aggiornare la configurazione dello stack.

2. Eseguire il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), che installa la versione corrente dei libri di ricette personalizzati nelle istanze dello stack, incluse le istanze online. Se è presente una versione precedente dei libri di cucina, verrà sovrascritta da questo comando.
3. Dopo aver aggiornato i libri di ricette personalizzati, eseguire la ricetta eseguendo il comando dello stack Execute Recipes (Esegui ricette) con l'opzione Recipes to execute (Ricette da eseguire) impostata su **`idcheck::default`**. Questo comando avvia un'esecuzione di Chef, con un elenco di esecuzione costituito dalla tua ricetta. Lasciare la pagina execute_recipes aperta.

Dopo che la ricetta è stata eseguita correttamente, è possibile verificarla esaminando il log di Chef per l'evento execute_recipes più recente. Nella pagina di esecuzione del comando execute_recipes, scegliere show (mostra) nella colonna Log dell'istanza iptest1 per visualizzare il log. Scorrere verso il basso per trovare il messaggio di log, che sarà simile al seguente.

```
...
[2015-05-13T20:03:47+00:00] INFO: Storing updated cookbooks/nodesearch/recipes/
default.rb in the cache.
[2015-05-13T20:03:47+00:00] INFO: Storing updated cookbooks/nodesearch/metadata.rb in
 the cache.
[2015-05-13T20:03:47+00:00] INFO: **********The instance ID is: 'i-8703b570'**********
[2015-05-13T20:03:47+00:00] INFO: Chef Run complete in 0.312518 seconds
...
```

Utilizzo di un libro di ricette esterno su un'istanza Linux: Berkshelf

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Berkshelf è disponibile solo per gli stack Linux Chef 11.10.

Prima di avviare l'implementazione di un libro di ricette, consulta la pagina relativa ai libri di ricette della community, che contiene i libri di ricette creati dai membri della community Chef per un'ampia gamma di scopi. Molti di questi libri di cucina possono essere utilizzati con AWS OpsWorks Stacks senza modifiche, quindi potresti sfruttarli per alcune delle tue attività invece di implementare tutto il codice da solo.

Per usare un libro di ricette esterno su un'istanza, devi installarlo e gestire eventuali dipendenze. L'approccio preferito consiste nell'implementare un libro di ricette che supporta un gestore delle dipendenze denominato Berkshelf. Berkshelf funziona su istanze Amazon, incluse EC2 le istanze AWS OpsWorks Stacks, ma è anche progettato per funzionare con Test Kitchen e Vagrant. Tuttavia, l'utilizzo su Vagrant è leggermente diverso rispetto a AWS OpsWorks Stacks, quindi questo argomento include esempi per entrambe le piattaforme. Per ulteriori informazioni su come usare Berkshelf, consulta Berkshelf.

Argomenti

- Utilizzo di Berkshelf con Test Kitchen e Vagrant
- Usare Berkshelf con Stacks AWS OpsWorks

Utilizzo di Berkshelf con Test Kitchen e Vagrant

Questo esempio illustra come utilizzare Berkshelf per installare il libro di ricette della community getting-started ed eseguirne la ricetta, che consente di installare un breve file di testo nella tua directory principale sull'istanza.

Per installare Berkshelf e inizializzare un libro di ricette

1. Installare il pacchetto Berkshelf nella workstation, come descritto di seguito.

   ```
   gem install berkshelf
   ```

   A seconda della workstation, questo comando può richiedere `sudo` oppure è possibile anche utilizzare un gestore di ambiente Ruby come RVM. Per verificare che Berkshelf sia stato installato correttamente, eseguire `berks --version`.

2. Il libro di ricette per questo argomento è denominato external_cookbook. È possibile utilizzare Berkshelf per creare un libro di ricette inizializzato anziché ricorrere all'approccio manuale degli argomenti precedenti. Per farlo, accedere alla directory `opsworks_cookbooks` ed eseguire il comando seguente.

   ```
   berks cookbook external_cookbook
   ```

   Il comando crea la directory `external_cookbook` e diverse sottodirectory standard di Chef e Test Kitchen, incluse `recipes` e `test`. Inoltre, crea versioni predefinite di numerosi file standard, inclusi i seguenti:

   - `metadata.rb`
   - I file di configurazione per Vagrant, Test Kitchen e Berkshelf
   - Una ricetta `default.rb` vuota nella directory `recipes`

   > **Note**
   >
   > Non è necessario eseguire `kitchen init`, il comando `berks cookbook` gestisce tali attività.

3. Esegui `kitchen converge`. Il libro di ricette appena creato non fa niente di interessante per il momento, ma converge.

> **Note**
>
> Inoltre puoi usare `berks init` per inizializzare un libro di ricette esistente al fine di utilizzare Berkshelf.

Per usare Berkshelf allo scopo di gestire le dipendenze esterne di un libro di ricette, la directory radice del libro di ricette deve contenere un `Berksfile`, ovvero un file di configurazione che specifica come Berkshelf dovrebbe gestire le dipendenze. Quando hai utilizzato `berks cookbook` per creare il libro di ricette `external_cookbook`, è stato creato un `Berksfile` con i seguenti contenuti.

```
source "https://supermarket.chef.io"
metadata
```

Il file include le dichiarazioni seguenti:

- `source`— L'URL di una fonte di libri di cucina.

  Un Berksfile può avere un qualsiasi numero di dichiarazioni `source`, ciascuna delle quali specifica un'origine predefinita per i libri di ricette dipendenti. Se non specifichi esplicitamente un'origine del libro di ricette, Berkshelf cerca nei repository predefiniti un libro di ricette con lo stesso nome. Il Berksfile predefinito include un singolo attributo `source` che specifica il repository del libro di ricette della community. Questo repository contiene il libro di ricette getting-started, che ti consente di lasciare la riga invariata.
- `metadata`— Indica a Berkshelf di includere le dipendenze dei libri di cucina dichiarate nel file del libro di cucina. `metadata.rb`

  Inoltre puoi dichiarare un libro di ricette dipendente nel Berksfile includendo un attributo `cookbook`, come illustrato in seguito.

Esistono due modi per dichiarare una dipendenza del libro di ricette:

- Includendo una dichiarazione `cookbook` nel Berksfile.

  Questo è l'approccio usato da Stacks. AWS OpsWorks Ad esempio per specificare il libro di ricette getting-started utilizzato in questo esempio, includi `cookbook "getting-started"` nel Berksfile. Berkshelf cercherà quindi nei repository predefiniti un libro di ricette con tale nome. Puoi inoltre utilizzare `cookbook` per specificare in modo esplicito un'origine del libro di ricette, nonché una determinata versione. Per ulteriori informazioni, consulta Berkshelf.
- Includendo una dichiarazione `metadata` nel Berksfile e dichiarando la dipendenza in `metadata.rb`.

  Questa dichiarazione indica a Berkshelf di includere le dipendenze del libro di ricette dichiarate in `metadata.rb`. Ad esempio, per dichiarare una dipendenza getting-started, aggiungere una dichiarazione `depends 'getting-started'` al file `metadata.rb` del libro di ricette.

Questo esempio utilizza il primo approccio, per garantire la coerenza con AWS OpsWorks Stacks.

Per installare il libro di ricette getting-started

1. Modificare il Berksfile predefinito per sostituire la dichiarazione `metadata` con una dichiarazione `cookbook` per `getting-started`. I contenuti devono essere simili ai seguenti.

```
source "https://supermarket.chef.io"

cookbook 'getting-started'
```

2. Eseguire `berks install`, che scarica il libro di ricette getting-started dal repository del libro di ricette della community alla directory Berkshelf della workstation, generalmente `~/.berkshelf`. Spesso questa directory si chiama semplicemente il Berkshelf. Cercare nella directory `cookbooks` del Berkshelf la directory per il libro di ricette getting-started, che avrà un nome simile a `getting-started-0.4.0`.
3. Sostituire `external_cookbook::default` nell'elenco di esecuzione `.kitchen.yml` con `getting-started::default`. In questo esempio non viene eseguita alcuna ricetta da external_cookbook. È fondamentalmente solo un modo per utilizzare il libro di ricette getting-started. Adesso il file `.kitchen.yml` avrà un aspetto simile al seguente.

```
---
driver:
  name: vagrant

provisioner:
  name: chef_solo

platforms:
  - name: ubuntu-12.04

suites:
  - name: default
    run_list:
      - recipe[getting-started::default]
    attributes:
```

4. Eseguire `kitchen converge` e utilizzare `kitchen login` per accedere all'istanza. La directory di accesso deve contenere un file denominato `chef-getting-started.txt` con elementi simili ai seguenti:

```
Welcome to Chef!

This is Chef version 11.12.8.
Running on ubuntu.
Version 12.04.
```

Test Kitchen installa libri di ricette nella directory `/tmp/kitchen/cookbooks` dell'istanza. Se si elencano i contenuti di tale directory, verranno visualizzati due libri di ricette: external_cookbook e getting-started.

5. Eseguire `kitchen destroy` per arrestare l'istanza. L'esempio successivo utilizza un' AWS OpsWorks istanza Stacks.

Usare Berkshelf con Stacks AWS OpsWorks

AWS OpsWorks Stacks supporta opzionalmente gli stack Berkshelf for Chef 11.10. Per usare Berkshelf con il tuo stack, devi procedere come descritto di seguito.

- Abilita Berkshelf per lo stack.

  AWS OpsWorks Stacks gestisce quindi i dettagli dell'installazione di Berkshelf sulle istanze dello stack.
- Aggiungi un Berksfile alla directory radice del repository del libro di ricette.

  Il Berksfile deve contenere le dichiarazioni `source` e `cookbook` per tutti i libri di ricette dipendenti.

Quando AWS OpsWorks Stacks installa il tuo repository di libri di cucina personalizzato su un'istanza, utilizza Berkshelf per installare i libri di cucina dipendenti dichiarati nel Berksfile del repository. Per ulteriori informazioni, consulta Utilizzo di Berkshelf.

Questo esempio mostra come utilizzare Berkshelf per installare il ricettario della community per iniziare su un'istanza Stacks. AWS OpsWorks Inoltre, installa una versione del libro di ricette personalizzato createfile, che crea un file in una directory specificata. Per ulteriori informazioni sul funzionamento di createfile, consulta Installazione di un file da un libro di ricette.

> **Note**
>
> Se è la prima volta che installi un ricettario personalizzato su uno AWS OpsWorks stack Stacks, dovresti prima seguire l'esempio. Esecuzione di una ricetta su un'istanza Linux

Inizia creando uno stack, come descritto brevemente di seguito. Per ulteriori informazioni, consulta Creare un nuovo stack.

Creare uno stack

1. Aprire la console AWS OpsWorks Stacks e fare clic su Add Stack (Aggiungi stack).
2. Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e fare clic su Add Stack (Aggiungi stack).

   - Nome — BerksTest
   - Chiave SSH predefinita: una coppia di EC2 chiavi Amazon

   Se devi creare una coppia di EC2 chiavi Amazon, consulta Amazon EC2 Key Pairs. La coppia di chiavi deve appartenere alla stessa regione AWS dell'istanza. L'esempio utilizza la regione predefinita degli Stati Uniti occidentali (Oregon).
3. Fare clic su Add a layer (Aggiungi un livello) e aggiungere un livello personalizzato allo stack con le impostazioni seguenti.

   - Nome: BerksTest
   - Nome abbreviato: berkstest

   Puoi effettivamente usare qualsiasi tipo di livello per questo esempio. Tuttavia, l'esempio non richiede nessuno dei pacchetti che vengono installati da altri tipi di livelli, pertanto un livello personalizzato è l'approccio più semplice.
4. Aggiungi un'istanza 24 ore su 24, 7 giorni su 7 al BerksTest layer con le impostazioni predefinite, ma non avviarla ancora.

Con AWS OpsWorks Stacks, i libri di cucina devono trovarsi in un archivio remoto con una struttura di directory standard. Quindi fornisci le informazioni di download a AWS OpsWorks Stacks, che scarica automaticamente il repository su ciascuna istanza dello stack all'avvio. Per semplicità, il repository

di questo esempio è un archivio Amazon S3 pubblico, AWS OpsWorks ma Stacks supporta anche archivi HTTP, repository Git e repository Subversion. Per ulteriori informazioni, consulta Repository dei libri di ricette.

I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Per creare il repository del libro di ricette

1. Nella directory `opsworks_cookbooks` creare una directory denominata `berkstest_cookbooks`. Se lo si preferisce, è possibile creare questa directory ovunque si ritiene che possa essere utile, perché verrà caricata in un repository.
2. Aggiungere un file denominato Berksfile a `berkstest_cookbooks` con i contenuti seguenti.

   ```
   source "https://supermarket.chef.io"

   cookbook 'getting-started'
   ```

   Questo file dichiara la dipendenza del libro di ricette getting-started e indica a Berkshelf di scaricarlo dal sito del libro di ricette della community.
3. Aggiungere una directory `createfile` a `berkstest_cookbooks` contenente quanto segue.

   - Un file `metadata.rb` con i seguenti contenuti.

     ```
     name "createfile"
     version "0.1.0"
     ```

   - Una directory `files/default` con un file `example_data.json` che include il contenuto seguente.

     ```
     {
       "my_name" : "myname",
       "your_name" : "yourname",
       "a_number" : 42,
       "a_boolean" : true
     }
     ```

Il nome e il contenuto del file sono arbitrari. La ricetta semplicemente copia il file nel percorso specificato.

- Una directory `recipes` contenente un file `default.rb` con il codice ricetta seguente.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end

cookbook_file "/srv/www/shared/example_data.json" do
  source "example_data.json"
  mode 0644
  action :create_if_missing
end
```

Questa ricetta crea `/srv/www/shared` e copia `example_data.json` in tale directory dalla directory `files` del libro di ricette.

4. Crea un `.zip` archivio di`berkstest_cookbooks`, carica l'archivio in un bucket Amazon S3, rendi l'archivio pubblico e registra l'URL dell'archivio.

Adesso puoi installare i libri di ricette ed eseguire la ricetta.

Per installare i libri di ricette ed eseguire le ricette

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: Http Archive
   - URL del repository: l'URL dell'archivio del libro di cucina che hai registrato in precedenza
   - Gestisci Berkshelf — Sì

   Le prime due impostazioni forniscono a AWS OpsWorks Stacks le informazioni necessarie per scaricare l'archivio dei libri di cucina nelle tue istanze. L'ultima impostazione abilita il supporto di

Berkshelf, che scarica il libro di ricette getting-started nell'istanza. Accettare i valori predefiniti per altre impostazioni e fare clic su Save (Salva) per aggiornare la configurazione dello stack.

2. Modifica il BerksTest layer per aggiungere le seguenti ricette all'evento Setup lifecycle del layer.

   - `getting-started::default`
   - `createfile::default`

3. Avviare l'istanza. L'evento Setup si verifica al termine dell'avvio dell'istanza. AWS OpsWorks Stacks installa quindi il repository di libri di cucina, utilizza Berkshelf per scaricare il ricettario introduttivo ed esegue la configurazione del layer e distribuisce le ricette, tra cui and. `getting-started::default` `createfile::default`
4. Una volta che l'istanza è online, utilizzare SSH per accedere. Verrà visualizzato un codice analogo al seguente

   - `/srv/www/shared` deve contenere `example_data.json`.
   - `/root` deve contenere `chef-getting-started.txt`.

     AWS OpsWorks Stacks esegue le ricette come root, quindi Getting-Started installa il file nella directory anziché nella directory home. `/root`

Utilizzo dell'SDK per Ruby: download di file da Amazon S3

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Alcune attività, come l'interazione con i servizi AWS, non possono essere gestite con le risorse Chef. Ad esempio, talvolta è preferibile archiviare file in remoto e scaricarli nell'istanza per mezzo di una ricetta. Puoi utilizzare la risorsa remote_file per scaricare file dai server remoti. Tuttavia, se desideri archiviare i tuoi file in un bucket Amazon S3, `remote_file` puoi scaricarli solo se l'ACL consente l'operazione.

Le ricette possono utilizzare l'AWS SDK per Ruby per accedere alla maggior parte dei servizi AWS. Questo argomento mostra come utilizzare l'SDK for Ruby per scaricare un file da un bucket S3.

> **Note**
>
> Per ulteriori informazioni su come utilizzare l'AWS SDK per Ruby per gestire la crittografia e la decrittografia, consulta AWS::S3::S3Object. I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Argomenti

- Utilizzo dell'SDK per Ruby su un'istanza Vagrant
- Utilizzo dell'SDK for Ruby AWS OpsWorks su un'istanza Stacks Linux
- Utilizzo dell'SDK for Ruby AWS OpsWorks su un'istanza Stacks Windows

Utilizzo dell'SDK per Ruby su un'istanza Vagrant

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo argomento descrive come una ricetta in esecuzione su un'istanza Vagrant può utilizzare AWS SDK per Rubyper scaricare un file da Amazon S3. Prima di iniziare, devi prima disporre di un set di AWS credenziali, una chiave di accesso e una chiave di accesso segreta, che consentano alla ricetta di accedere ad Amazon S3.

> **Important**
>
> Ti consigliamo di non utilizzare credenziali dell'account root a questo scopo. Invece, crea un utente con una politica appropriata e fornisci quelle credenziali alla ricetta.

Fai attenzione a non inserire le credenziali, nemmeno le credenziali degli utenti IAM, in una posizione accessibile al pubblico, ad esempio caricando un file contenente le credenziali in un repository pubblico o Bitbucket. GitHub Questo espone le tue credenziali e può compromettere la sicurezza del tuo account.
Le ricette eseguite su un' EC2 istanza EC2 Amazon possono utilizzare un approccio ancora migliore, un ruolo IAM, come descritto inUtilizzo dell'SDK for Ruby AWS OpsWorks su un'istanza Stacks Linux.
I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Se non disponi già di un utente appropriato, puoi crearne uno nel modo seguente. Per ulteriori informazioni, consulta Cos'è IAM.

⚠ Warning

Gli utenti IAM dispongono di credenziali a lungo termine, il che rappresenta un rischio per la sicurezza. Per ridurre questo rischio, si consiglia di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari.

Per creare un utente IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Utenti e, se necessario, scegli Aggiungi utenti per creare un nuovo utente amministrativo.
3. Nella pagina Imposta autorizzazioni, scegli Allega direttamente le politiche.
4. Digita **S3** nella casella di ricerca Politiche di autorizzazione per visualizzare le politiche di Amazon S3.

   Scegli AmazonS3. ReadOnlyAccess Se preferisci, puoi specificare una politica che conceda autorizzazioni più ampie, come AmazonS3 FullAccess, ma la prassi standard prevede di concedere solo le autorizzazioni necessarie. In questo caso, la ricetta non farà che scaricare un file, pertanto un accesso in sola lettura è sufficiente.

5. Scegli Next (Successivo).
6. Scegli Crea utente
7. Quindi crea le chiavi di accesso per il tuo utente. Per ulteriori informazioni sulla creazione di chiavi di accesso, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM.

A questo punto, devi fornire un file da scaricare. Questo esempio presuppone che inserirai un file denominato `myfile.txt` in un nuovo bucket S3 chiamato `cookbook_bucket`.

Per fornire un file per il download

1. Creare un file denominato `myfile.txt` con il testo seguente e salvarlo in una posizione comoda sulla workstation.

```
This is the file that you just downloaded from Amazon S3.
```

2. Sulla console Amazon S3, crea un bucket denominato `cookbook_bucket` nella regione Standard e caricalo `myfile.txt` nel bucket.

Configurare il libro di ricette come segue.

Per configurare il libro di ricette

1. Creare una directory in `opsworks_cookbooks` denominata `s3bucket` e accedervi.
2. Inizializzare e configurare Test Kitchen, come descritto in Esempio 1: installazione di pacchetti.
3. Sostituire il testo in `.kitchen.yml` con il seguente.

```
---
driver:
  name: vagrant

provisioner:
  name: chef_solo
  environments_path: ./environments

platforms:
  - name: ubuntu-14.04
```

```
suites:
  - name: s3bucket
    provisioner:
      solo_rb:
        environment: test
    run_list:
      - recipe[s3bucket::default]
    attributes:
```

4. Aggiungere due directory a `s3bucket`: `recipes` ed `environments`.
5. Crea un file di ambiente denominato `test.json` `default_attributes` nella sezione seguente, sostituendo `secret_key` i valori `access_key` and con le chiavi corrispondenti per il tuo utente. Salvare il file nella cartella `environments` del libro di ricette.

```
{
  "default_attributes" : {
    "cookbooks_101" : {
      "access_key": "AKIAIOSFODNN7EXAMPLE",
      "secret_key" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

Hai a disposizione diversi metodi per fornire le credenziali a una ricetta in esecuzione su un'istanza. Il concetto fondamentale è limitare le possibilità di esporre accidentalmente le chiavi e compromettere la sicurezza del tuo account. Per questo motivo, non è consigliabile utilizzare valori chiave espliciti nel tuo codice. L'esempio invece colloca i valori chiave nell'oggetto nodo, che consente alla ricetta di farvi riferimento utilizzando la sintassi del nodo anziché esponendo valori letterali. Devi disporre di privilegi root per accedere all'oggetto nodo, che limita le possibilità che le chiavi possano essere esposte. Per ulteriori informazioni, consulta Best practice per la gestione delle chiavi di accesso AWS.

> **Note**
>
> L'esempio utilizza gli attributi nidificati, con `cookbooks_101` come primo elemento. Questa pratica limita la possibilità di un conflitto di nomi se vi sono altri attributi `access_key` o `secret_key` nell'oggetto nodo.

La ricetta seguente scarica `myfile.text` dal bucket `cookbook_bucket`.

```
gem_package "aws-sdk ~> 3" do
  action :install
end

ruby_block "download-object" do
  block do
    require 'aws-sdk'

    s3 = Aws::S3::Client.new(
          :access_key_id => "#{node['cookbooks_101']['access_key']}",
          :secret_access_key => "#{node['cookbooks_101']['secret_key']}")

    myfile = s3.bucket['cookbook_bucket'].objects['myfile.txt']
    Dir.chdir("/tmp")
    File.open("myfile.txt", "w") do |f|
      f.write(myfile.read)
      f.close
    end
  end
  action :run
end
```

La prima parte della ricetta installa l'SDK for Ruby, che è un pacchetto gem. La risorsa gem_package installa gem che verranno utilizzati da ricette o altre applicazioni.

> **Note**
>
> Di norma la tua istanza ha due istanze di Ruby, che sono generalmente versioni diverse. Una è un'istanza dedicata che viene utilizzata dal client Chef. L'altra viene utilizzato da applicazioni e ricette in esecuzione sull'istanza. È importante comprendere questa distinzione quando si installano pacchetti gem, perché per installare gem sono disponibili due risorse, gem_package e chef_gem. Se le applicazioni o le ricette utilizzano il pacchetto gem, installarlo con `gem_package`. `chef_gem` è solo per i pacchetti gem utilizzati dal client Chef.

Il resto della ricetta è una risorsa ruby_block, contenente il codice Ruby che scarica il file. Potresti pensare che, poiché una ricetta è un'applicazione Ruby, puoi inserire il codice direttamente nella ricetta. Tuttavia, un'esecuzione di Chef compila tutto questo codice prima dell'esecuzione di

qualunque risorsa. Se inserisci il codice di esempio direttamente nella ricetta, Ruby cercherà di risolvere l'istruzione `require 'aws-sdk'` prima di eseguire la risorsa `gem_package`. Poiché l'SDK for Ruby non è ancora stato installato, la compilazione avrà esito negativo.

Il codice in una risorsa `ruby_block` non viene compilato finché tale risorsa non viene eseguita. In questo esempio, la `ruby_block` risorsa viene eseguita dopo che la `gem_package` risorsa ha terminato l'installazione dell'SDK for Ruby, quindi il codice verrà eseguito correttamente.

Il codice in `ruby_block` funziona come segue.

1. Crea un nuovo oggetto `Aws::S3`, che fornisce l'interfaccia di servizio.

   Le chiavi segrete e di accesso vengono specificate facendo riferimento ai valori memorizzati nell'oggetto nodo.
2. Chiama l'associazione `bucket.objects` dell'oggetto `S3`, che restituisce un oggetto `Aws::S3::Object` denominato `myfile` che rappresenta `myfile.txt`.
3. Utilizza `Dir.chdir` per impostare la directory di lavoro su `/tmp`.
4. Apre un file denominato `myfile.txt`, scrive i contenuti di `myfile` nel file e chiude il file.

Per eseguire la ricetta

1. Creare un file denominato `default.rb` con la ricetta di esempio e salvarlo nella directory `recipes`.
2. Esegui `kitchen converge`.
3. Eseguire `kitchen login` per accedere all'istanza, quindi eseguire `ls /tmp`. Dovrebbe essere possibile vedere `myfile.txt`, insieme a diversi file e directory di Test Kitchen.

   ```
   vagrant@s3bucket-ubuntu-1204:~$ ls /tmp
   install.sh  kitchen  myfile.txt  stderr
   ```

   È inoltre possibile eseguire `cat /tmp/myfile.txt` per verificare che il contenuto del file sia corretto.

Al termine dell'operazione, eseguire `kitchen destroy` per terminare l'istanza.

Utilizzo dell'SDK for Ruby AWS OpsWorks su un'istanza Stacks Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo argomento descrive come utilizzare l'SDK for Ruby AWS OpsWorks su un'istanza Stacks Linux per scaricare un file da un bucket Amazon S3. AWS OpsWorks Stacks installa automaticamente l'SDK for Ruby su ogni istanza Linux. Tuttavia, al momento della creazione di un oggetto client del servizio, devi fornire un set di credenziali di AWS idoneo `AWS::S3.new` o l'equivalente per altri servizi.

I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Utilizzo dell'SDK per Ruby su un'istanza Vagrant mostra come mitigare il rischio di esposizione delle tue credenziali archiviando le credenziali nell'oggetto nodo e facendo riferimento agli attributi nel codice della tua ricetta. Quando esegui ricette su un' EC2istanza Amazon, hai un'opzione ancora migliore, un ruolo IAM.

Un ruolo IAM funziona in modo molto simile a un utente IAM. Ha una policy collegata che concede autorizzazioni per l'utilizzo di vari servizi AWS. Tuttavia, assegni un ruolo a un' EC2istanza Amazon anziché a un individuo. Le applicazioni in esecuzione su tale istanza possono quindi acquisire le autorizzazioni concesse dalla policy collegata. Con un ruolo, le credenziali non compaiono mai nel codice, neanche indirettamente. Questo argomento descrive come utilizzare un ruolo IAM per eseguire la ricetta da Utilizzo dell'SDK per Ruby su un'istanza Vagrant un' EC2 istanza Amazon.

Puoi eseguire questa ricetta con Test Kitchen utilizzando il driver kitchen-ec2, come descritto in Esempio 9: utilizzo di Amazon EC2 Instances. Tuttavia, l'installazione dell'SDK for Ruby EC2 sulle istanze Amazon è piuttosto complicata e non è qualcosa di cui devi preoccuparti per Stacks. AWS OpsWorks Tutte le istanze AWS OpsWorks Stacks Linux hanno l'SDK for Ruby installato per impostazione predefinita. Per semplicità, l'esempio utilizza quindi un'istanza Stacks. AWS OpsWorks

Il primo passaggio consiste nell'impostare il ruolo IAM. Questo esempio utilizza l'approccio più semplice, che consiste nell'utilizzare il EC2 ruolo Amazon creato da AWS OpsWorks Stacks quando crei il primo stack. È denominato `aws-opsworks-ec2-role`. Tuttavia, AWS OpsWorks Stacks non assegna una policy a quel ruolo, quindi per impostazione predefinita non concede autorizzazioni.

È necessario allegare la `AmazonS3ReadOnlyAccess` politica al `aws-opsworks-ec2-role` ruolo per concedere le autorizzazioni appropriate. Per ulteriori informazioni su come allegare una policy a un ruolo, consulta Adding IAM identity permissions (console) nella IAM User Guide.

Specifica il ruolo durante la creazione o l'aggiornamento di uno stack. Configura uno stack con un livello personalizzato, come descritto in Esecuzione di una ricetta su un'istanza Linux, con un'aggiunta. Nella pagina Add Stack, verifica che il profilo predefinito dell'istanza IAM sia impostato su 2 ruoli. aws-opsworks-ec AWS OpsWorks Stacks assegnerà quindi quel ruolo a tutte le istanze dello stack.

La procedura per la configurazione del libro di ricette è simile a quella utilizzata da Esecuzione di una ricetta su un'istanza Linux. Di seguito è riportato un breve riepilogo. Per i dettagli, fai riferimento a tale esempio.

Per configurare il libro di ricette

1. Creare una directory denominata `s3bucket_ops` e accedervi.
2. Creare un file `metadata.rb` con il seguente contenuto e salvarlo in `s3bucket_ops`.

```
name "s3bucket_ops"
version "0.1.0"
```

3. Creare una directory `recipes` in `s3bucket_ops`.
4. Creare un file `default.rb` con la seguente ricetta e salvarlo nella directory `recipes`.

```
Chef::Log.info("******Downloading a file from Amazon S3.******")

ruby_block "download-object" do
  block do
    require 'aws-sdk'

    s3 = AWS::S3.new

    myfile = s3.buckets['cookbook_bucket'].objects['myfile.txt']
```

```
    Dir.chdir("/tmp")
    File.open("myfile.txt", "w") do |f|
      f.syswrite(myfile.read)
      f.close
    end
  end
  action :run
end
```

5. Crea un `.zip` archivio di `s3bucket_ops` e caricalo in un bucket Amazon S3. Per semplicità, rendere pubblico l'archivio, quindi registrare l'URL dell'archivio per utilizzarlo in futuro. Puoi anche archiviare i tuoi libri di cucina in un archivio privato di Amazon S3 o in diversi altri tipi di repository. Per ulteriori informazioni, consulta Repository dei libri di ricette.

Questa ricetta è simile a quella utilizzata dall'esempio precedente, con le seguenti eccezioni.

- Poiché AWS OpsWorks Stacks ha già installato l'SDK for Ruby, la risorsa è stata `chef_gem` eliminata.
- La ricetta non passa credenziali ad `AWS::S3.new`.

  Le credenziali sono assegnate automaticamente all'applicazione in base al ruolo dell'istanza.
- La ricetta impiega `Chef::Log.info` per aggiungere un messaggio al log di Chef.

Crea uno stack per questo esempio nel modo seguente. Puoi anche utilizzare uno stack Windows esistente. È sufficiente aggiornare i libri di ricette, come descritto di seguito.

Per creare uno stack

1. Aprire la console AWS OpsWorks Stacks e fare clic su Add Stack (Aggiungi stack).
2. Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e fare clic su Add Stack (Aggiungi stack).

   - Nome: RubySDK
   - Chiave SSH predefinita: una coppia di EC2 chiavi Amazon

   Se devi creare una coppia di EC2 chiavi Amazon, consulta Amazon EC2 Key Pairs. La coppia di chiavi deve appartenere alla stessa regione AWS dell'istanza. L'esempio utilizza la regione predefinita degli Stati Uniti occidentali (Oregon).

3. Fare clic su Add a layer (Aggiungi un livello) e aggiungere un livello personalizzato allo stack con le impostazioni seguenti.

   - Nome: S3Download
   - Nome breve: s3download

   Per gli stack Linux sarà appropriato qualunque tipo di livello, tuttavia l'esempio non richiede nessuno dei pacchetti che vengono installati da altri tipi di livello, pertanto un livello personalizzato è l'approccio più semplice.
4. Aggiungere un'istanza 24/7 con impostazioni predefinite al livello e avviarla.

Ora puoi installare ed eseguire la ricetta

Per eseguire la ricetta

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: Http Archive
   - URL del repository: l'URL di archivio del libro di cucina che hai registrato in precedenza.

   Utilizzare i valori predefiniti per altre impostazioni e fare clic su Save (Salva) per aggiornare la configurazione dello stack.
2. Eseguire il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), che installa la versione corrente dei libri di ricette personalizzati nelle istanze dello stack. Se è presente una versione precedente dei libri di cucina, verrà sovrascritta da questo comando.
3. Eseguire la ricetta utilizzando il comando dello stack Execute Recipes (Esegui ricette) con Recipes to execute (Ricette da eseguire) impostato su **`s3bucket_ops::default`**. Questo comando avvia un'esecuzione di Chef, con un elenco di esecuzione costituito da `s3bucket_ops::default`.

   > **Note**
   >
   > In genere, fai in modo che AWS OpsWorks Stacks esegua le tue ricette automaticamente assegnandole all'evento del ciclo di vita appropriato. Puoi eseguire

tali ricette attivando manualmente l'evento e Puoi utilizzare un comando dello stack per attivare gli eventi Setup e Configure e un comando di distribuzione per attivare gli eventi Deploy e Undeploy.

Dopo aver eseguito correttamente la ricetta, puoi verificarla.

Per verificare s3bucket_ops

1. Il primo passaggio consiste nell'esaminare il log di Chef. Lo stack deve disporre di un'istanza denominata opstest1. Nella pagina Instances (Istanze) fare clic su show (mostra) nella colonna Log dell'istanza per visualizzare il log di Chef. Scorrere verso il basso per trovare il messaggio di log.

```
...
[2014-07-31T17:01:45+00:00] INFO: Storing updated cookbooks/opsworks_cleanup/
attributes/customize.rb in the cache.
[2014-07-31T17:01:45+00:00] INFO: Storing updated cookbooks/opsworks_cleanup/
metadata.rb in the cache.
[2014-07-31T17:01:46+00:00] INFO: ******Downloading a file from Amazon S3.******
[2014-07-31T17:01:46+00:00] INFO: Processing template[/etc/hosts] action create
 (opsworks_stack_state_sync::hosts line 3)
...
```

2. Utilizzare SSH per accedere all'istanza ed elencare i contenuti di `/tmp`.

Utilizzo dell'SDK for Ruby AWS OpsWorks su un'istanza Stacks Windows

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo esempio presuppone che tu abbia già completato l'esempio Esecuzione di una ricetta su un'istanza Windows. In caso contrario, ti consigliamo di iniziare da questo esempio, che descrive in particolare come abilitare l'accesso RDP alle istanze.
> I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Questo argomento descrive come utilizzare AWS SDK per Rubysu un'istanza Windows AWS OpsWorks Stacks per scaricare un file da un bucket S3.

Se un'applicazione Ruby ha l'esigenza di accedere a una risorsa AWS, devi fornirle un set di credenziali di AWS con le autorizzazioni appropriate. Per le ricette, l'opzione migliore per fornire le credenziali AWS è utilizzare un ruolo AWS Identity and Access Management (IAM). Un ruolo IAM funziona in modo molto simile a un utente IAM: ha una policy allegata che concede le autorizzazioni per utilizzare i vari servizi. AWS Tuttavia, assegni un ruolo a un'istanza Amazon Elastic Compute Cloud EC2 (Amazon) anziché a un individuo. Le applicazioni in esecuzione su tale istanza possono quindi acquisire le autorizzazioni concesse dalla policy collegata. Con un ruolo, le credenziali non compaiono mai nel codice, neanche indirettamente.

Il primo passo è configurare il ruolo IAM. Questo esempio utilizza l'approccio più semplice, che consiste nell'utilizzare il EC2 ruolo Amazon creato da AWS OpsWorks Stacks quando crei il primo stack. È denominato `aws-opsworks-ec2-role`. Tuttavia, AWS OpsWorks Stacks non assegna una policy a quel ruolo, quindi per impostazione predefinita non concede autorizzazioni.

È necessario allegare la `AmazonS3ReadOnlyAccess` politica al `aws-opsworks-ec2-role` ruolo per concedere le autorizzazioni appropriate. Per ulteriori informazioni su come allegare una policy a un ruolo, consulta Adding IAM identity permissions (console) nella IAM User Guide.

Specifica il ruolo durante la creazione o l'aggiornamento di uno stack. Configura uno stack con un livello personalizzato, come descritto in Esecuzione di una ricetta su un'istanza Windows, con un'aggiunta. Nella pagina Add Stack, verifica che il profilo predefinito dell'istanza IAM sia impostato su 2 ruoli. aws-opsworks-ec AWS OpsWorks Stacks assegnerà quindi quel ruolo a tutte le istanze dello stack.

La procedura per la configurazione del libro di ricette è simile a quella utilizzata da Esecuzione di una ricetta su un'istanza Linux. Di seguito è riportato un breve riepilogo. Per i dettagli, fai riferimento a tale esempio.

Per configurare il libro di ricette

1. Creare una directory denominata `s3bucket_ops` e accedervi.
2. Creare un file `metadata.rb` con il seguente contenuto e salvarlo in `s3bucket_ops`.

```
name "s3download"
version "0.1.0"
```

3. Creare una directory `recipes` in `s3download`.
4. Creare un file `default.rb` con la seguente ricetta e salvarlo nella directory `recipes`. Sostituiscilo *`windows-cookbooks`* con il nome del bucket S3 che utilizzerai per archiviare il file da scaricare.

```
Chef::Log.info("******Downloading an object from S3******")

chef_gem "aws-sdk-s3" do
  compile_time false
  action :install
end

ruby_block "download-object" do
  block do
    require 'aws-sdk-s3'

    Aws.use_bundled_cert!

    s3_client = Aws::S3::Client.new(region:'us-west-2')

    s3_client.get_object(bucket: 'windows-cookbooks',
                         key: 'myfile.txt',
                         response_target: '/chef/myfile.txt')
  end
  action :run
end
```

5. Creare un archivio `.zip` di `s3download` e caricare il file in un bucket S3. Rendere il file pubblico e registrare l'URL per utilizzarlo successivamente.
6. Creare un file di testo denominato `myfile.txt` e caricarlo in un bucket S3. Questo è il file che verrà scaricato dalla ricetta, perciò è possibile usare qualsiasi bucket comodo.

La ricetta esegue le seguenti attività.

1: Installa l'SDK per Ruby v2.

L'esempio utilizza l'SDK for Ruby per scaricare l'oggetto. Tuttavia, AWS OpsWorks Stacks non installa questo SDK su istanze Windows, quindi la prima parte della ricetta utilizza una `chef_gem`risorsa per gestire tale attività. Questa risorsa viene utilizzata per installare gem per l'utilizzo da parte di Chef, che include ricette.

2: Scarica il file.

La terza parte della ricetta utilizza una `ruby_block`risorsa per eseguire il codice SDK for Ruby v2 da `myfile.txt` scaricare da un *`windows-cookbooks`* bucket S3 denominato nella directory dell'istanza. `/chef` Sostituisci *`windows-cookbooks`* con il nome del bucket che contiene `myfile.txt`.

> **Note**
>
> Una ricetta è un'applicazione Ruby, pertanto puoi inserire il codice Ruby nel corpo della ricetta. Tale codice non deve trovarsi in una risorsa `ruby_block`. Tuttavia, Chef esegue prima il codice Ruby nel corpo della ricetta, quindi ciascuna risorsa, nell'ordine. Per questo esempio, se inserisci il codice di download nel corpo della ricetta, fallirà perché dipende dall'SDK for Ruby `chef_gem` e la risorsa che installa l'SDK non è ancora stata eseguita. Il codice nella `ruby_block` risorsa viene eseguito quando viene eseguita la risorsa e ciò avviene dopo che la `chef_gem` risorsa ha installato l'SDK for Ruby.

Crea uno stack per questo esempio nel modo seguente. Puoi anche utilizzare uno stack Windows esistente. È sufficiente aggiornare i libri di ricette, come descritto di seguito.

Creare uno stack

1. Aprire la console AWS OpsWorks Stacks e scegliere Add Stack (Aggiungi stack). Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e scegliere Add Stack (Aggiungi stack).

   - Nome: S3Download
   - Regione: Stati Uniti occidentali (Oregon)

     Questo esempio funzionerà in qualsiasi regione, ma consigliamo di utilizzare Stati Uniti occidentali (Oregon) per i tutorial.
   - Sistema operativo predefinito: Microsoft Windows Server 2012 R2
2. Scegliere Add a layer (Aggiungi un livello) e aggiungere un livello personalizzato allo stack con le impostazioni seguenti.

   - Nome: S3Download
   - Nome breve: s3download
3. Aggiungere un'istanza 24/7 con impostazioni predefinite al livello S3Download e avviarla.

Ora puoi installare ed eseguire la ricetta

Per eseguire la ricetta

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: S3 Archive.
   - URL del repository: l'URL di archivio del libro di cucina che hai registrato in precedenza.

   Accettare i valori predefiniti per altre impostazioni e scegliere Save (Salva) per aggiornare la configurazione dello stack.
2. Eseguire il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), che installa la versione più recente del libro di ricette personalizzato nelle istanze online dello stack. Se è presente una versione precedente dei libri di cucina, verrà sovrascritta da questo comando.
3. Eseguire la ricetta utilizzando il comando dello stack Execute Recipes (Esegui ricette) con Recipes to execute (Ricette da eseguire) impostato su **`s3download::default`**.

Questo comando avvia un'esecuzione di Chef, con un elenco di esecuzione costituito da `s3download::default`.

> **Note**
>
> In genere, fai in modo che AWS OpsWorks Stacks esegua le tue ricette automaticamente assegnandole all'evento del ciclo di vita appropriato. Puoi anche eseguire tali ricette attivando manualmente l'evento. Puoi utilizzare un comando dello stack per attivare gli eventi Setup e Configure e un comando di distribuzione per attivare gli eventi Deploy e Undeploy.

Dopo aver eseguito correttamente la ricetta, puoi verificarla.

Per verificare s3download

1. Il primo passaggio consiste nell'esaminare il log di Chef. Lo stack deve disporre di un'istanza denominata s3download1. Nella pagina Instances (Istanze) scegliere show (mostra) nella colonna Log dell'istanza per visualizzare il log di Chef. Scorrere verso il basso per trovare il messaggio di log.

```
...
[2015-05-01T21:11:04+00:00] INFO: Loading cookbooks [s3download@0.0.0]
[2015-05-01T21:11:04+00:00] INFO: Storing updated cookbooks/s3download/recipes/
default.rb in the cache.
[2015-05-01T21:11:04+00:00] INFO: ******Downloading an object from S3******
[2015-05-01T21:11:04+00:00] INFO: Processing chef_gem[aws-sdk] action install
 (s3download::default line 3)
[2015-05-01T21:11:05+00:00] INFO: Processing ruby_block[download-object] action run
 (s3download::default line 8)
...
```

2. Utilizzare RDP per accedere all'istanza ed esaminare i contenuti di `c:\chef`.

Installazione di software Windows

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi esempi presuppongono che tu abbia già completato l'esempio Esecuzione di una ricetta su un'istanza Windows. In caso contrario, ti consigliamo di iniziare da questo esempio, che descrive in particolare come abilitare l'accesso RDP alle istanze.

Le istanze di Windows iniziano con Windows Server 2012 R2 Standard, pertanto devi installare software. I dettagli variano a seconda del tipo di software.

- Le funzionalità di Windows sono componenti di sistema opzionali, tra cui i framework.NET e Internet Information Services (IIS), che puoi scaricare sulla tua istanza.
- Di solito, il software di terze parti viene fornito in un pacchetto di installazione, come un file MSI, che devi scaricare nell'istanza ed eseguire.

  Anche determinato software Microsoft viene fornito in un pacchetto di installazione.

In questa sezione viene descritto come implementare libri di ricette per installare pacchetti e funzionalità Windows. Inoltre, introduce il libro di ricette Windows di Chef, che contiene risorse e funzioni dell'helper che semplificano l'implementazione di ricette per le istanze Windows.

Argomenti

- Installazione di una funzionalità di Windows: IIS
- Installazione di un pacchetto in un'istanza Windows

Installazione di una funzionalità di Windows: IIS

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le funzionalità di Windows sono un insieme di componenti di sistema opzionali, tra cui i framework.NET e Internet Information Services (IIS). Questo argomento descrive come implementare un ricettario per installare una funzionalità di uso comune, Internet Information Services (IIS).

> **Note**
>
> Installazione di un pacchetto mostra come installare software fornito in un pacchetto di installazione, come un file MSI, che devi scaricare nell'istanza ed eseguire. Libri di ricette IIS

Esecuzione di una ricetta su un'istanza Windows mostra come utilizzare una risorsa `powershell_script` per installare una funzionalità di Windows. Questo esempio mostra un approccio alternativo: usa la risorsa del ricettario Chef Windows. `windows_feature` Questo libro di ricette contiene un set di risorse che utilizzano Gestione e manutenzione immagini distribuzione per eseguire un'ampia gamma di attività su Windows, inclusa l'installazione di funzionalità.

> **Note**
>
> Chef include inoltre un libro di ricette IIS, che puoi utilizzare per gestire IIS. Per ulteriori informazioni, consulta la sezione relativa al libro di ricette IIS.

Per configurare il libro di ricette

1. Vai all' GitHub archivio dei libri di cucina di Windows e scarica il ricettario. `windows`

   Questo esempio presuppone che il repository `windows` verrà scaricato come file .zip, ma è anche possibile clonare il repository se lo si preferisce.

2. Vai al repository dei libri di cucina chef_handler e scarica il ricettario. GitHub `chef-handler`

   Il libro di ricette `windows` dipende da `chef_handler` e non può essere utilizzato direttamente. Questo esempio presuppone che il repository `chef_handler` verrà scaricato come file .zip, ma è anche possibile clonare il repository se lo si preferisce.
3. Estrarre i libri di ricette `windows` e `chef_handler` nelle directory nella directory dei libri di ricette denominati, rispettivamente, `windows` e `chef_handler`.
4. Creare una directory nella directory dei libri di ricette denominata `install-iis` e accedervi.
5. Aggiungere un file `metadata.rb` a `install-iis` con il contenuto seguente.

   ```
   name "install-iis"
   version "0.1.0"

   depends "windows"
   ```

   La direttiva `depends` consente di utilizzare le risorse del libro di ricette `windows` nelle ricette.
6. Aggiungere una directory `recipes` a `install-iis` e aggiungere un file denominato `default.rb` a tale directory che contiene il codice della ricetta seguente.

   ```
   %w{ IIS-WebServerRole IIS-WebServer }.each do |feature|
     windows_feature feature do
       action :install
     end
   end

   service 'w3svc' do
     action [:start, :enable]
   end
   ```

   La ricetta utilizza la risorsa `windows_feature` del libro di ricette `windows` per installare gli elementi seguenti:

   1. Il ruolo Server Web IIS.
   2. Il Server Web IIS.

   La ricetta usa quindi una risorsa `service` per avviare e abilitare il servizio IIS (W3SVC).

> **Note**
>
> Per un elenco completo delle funzionalità di Windows disponibili, utilizzare RDP per accedere all'istanza, aprire una finestra del prompt dei comandi ed eseguire il comando seguente. L'elenco è abbastanza lungo.
>
> ```
> dism /online /Get-Features
> ```

7. Creare un archivio `.zip` che contiene i libri di ricette `install-iis`, `chef_handler` e `windows` e caricare l'archivio in un bucket S3. Rendere pubblico l'archivio e registrare l'URL per utilizzarlo successivamente. Questo esempio presuppone che l'archivio sia denominato `install-iis.zip`. Per ulteriori informazioni, consulta Repository dei libri di ricette.

   I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Crea uno stack per questo esempio nel modo seguente. Puoi anche utilizzare uno stack Windows esistente. È sufficiente aggiornare i libri di ricette, come descritto di seguito.

Creare uno stack

1. Aprire la console AWS OpsWorks Stacks e scegliere Add Stack (Aggiungi stack). Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e scegliere Add Stack (Aggiungi stack).

   - Nome: installIIS
   - Regione: Stati Uniti occidentali (Oregon)

     Questo esempio funzionerà in qualsiasi regione, ma consigliamo di utilizzare Stati Uniti occidentali (Oregon) per i tutorial.
   - Sistema operativo predefinito: Microsoft Windows Server 2012 R2
2. Scegliere Add a layer (Aggiungi un livello) e aggiungere un livello personalizzato allo stack con le impostazioni seguenti.

   - Nome: IIS

- Nome abbreviato: iis

3. Aggiungere un'istanza 24/7 con impostazioni predefinite al livello IIS e avviarla.

Adesso puoi installare il libro di ricette ed eseguire la ricetta

Per installare il libro di ricette ed eseguire la ricetta

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: S3 Archive
   - URL del repository: l'URL dell'archivio dei libri di cucina che hai registrato in precedenza.

   Accettare i valori predefiniti per altre impostazioni e scegliere Save (Salva) per aggiornare la configurazione dello stack.

2. Eseguire il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), che installa la versione più recente dei libri di ricette personalizzati nelle istanze online dello stack. Se è presente una versione precedente dei libri di cucina, verrà sovrascritta da questo comando.

3. Eseguire la ricetta utilizzando il comando dello stack Execute Recipes (Esegui ricette) con Recipes to execute (Ricette da eseguire) impostato su **`install-iis::default`**. Questo comando avvia un'esecuzione di Chef, che esegue le ricette specificate.

   > **Note**
   >
   > Questo esempio utilizza Execute Recipes per comodità, ma in genere AWS OpsWorks Stacks esegue le ricette automaticamente assegnandole all'evento del ciclo di vita appropriato. Puoi eseguire tali ricette attivando manualmente l'evento e Puoi utilizzare un comando dello stack per attivare gli eventi Setup e Configure e un comando di distribuzione per attivare gli eventi Deploy e Undeploy.

4. Per verificare l'installazione, utilizzare RDP per connettersi all'istanza e aprire Esplora risorse. Il file system deve avere una directory `C:\inetpub`. Se si verifica l'elenco dei servizi nell'applicazione del pannello di controllo degli strumenti di amministrazione, IIS deve essere quasi in fondo. Tuttavia, sarà denominato World Wide Web Publishing Service, non IIS.

Installazione di un pacchetto in un'istanza Windows

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo esempio presuppone che tu abbia già completato l'esempio Esecuzione di una ricetta su un'istanza Windows. In caso contrario, ti consigliamo di iniziare da questo esempio, che descrive in particolare come abilitare l'accesso RDP alle istanze.

Se il tuo software viene fornito in un pacchetto di installazione, come un file MSI, devi scaricare il file nell'istanza ed eseguirlo. Questo esempio illustra come implementare un libro di ricette per installare un pacchetto MSI, il runtime Python, nonché come definire variabili di ambiente associate. Per ulteriori informazioni su come installare le funzionalità di Windows, ad esempio IIS, consulta Installazione di una funzionalità di Windows: IIS.

Per configurare il libro di ricette

1. Creare una directory denominata `installpython` e accedervi.
2. Aggiungere un file `metadata.rb` a `installpython` con il contenuto seguente.

   ```
   name "installpython"
   version "0.1.0"
   ```

3. Aggiungere le directory `recipes` e `files` a `installpython` e aggiungere una directory `default` ai file.
4. Scaricare un pacchetto Python da Versioni di Python per Windows nella directory `files\default` del libro di ricette. Questo esempio installa la versione Windows

x86-64 di Python 3.5.0a3, che utilizza un programma di installazione MSI denominato `python-3.4.3.amd64.msi`.

5. Aggiungere un file denominato `default.rb` alla directory `recipes` con il seguente codice ricetta.

```
directory 'C:\tmp' do
  rights :full_control, 'Everyone'
  recursive true
  action :create
end

cookbook_file 'C:\tmp\python-3.4.3.amd64.msi' do
  source "python-3.4.3.amd64.msi"
  rights :full_control, 'Everyone'
  action :create
end

windows_package 'python' do
  source 'C:\tmp\python-3.4.3.amd64.msi'
  action :install
end

env "PATH" do
  value 'c:\python34'
  delim ";"
  action :modify
end
```

La ricetta effettua le seguenti operazioni:

1. Utilizzare una risorsa directory per creare una directory `C:\tmp`.

   Per ulteriori informazioni su questa risorsa, consulta Esempio 3: creazione di directory.

2. Utilizza una risorsa cookbook_file per copiare il programma di installazione dalla directory `files\default` del libro di ricette in `C:\tmp`.

   Per ulteriori informazioni su questa risorsa, consulta Installazione di un file da un libro di ricette.

3. Utilizza una risorsa windows_package per eseguire il programma di installazione MSI, che installa Python in `c:\python34`.

Il programma di installazione crea le directory richieste e installa i file, ma non modifica la variabile di ambiente PATH del sistema.

4. Utilizza una risorsa env per aggiungere `c:\python34` al percorso di sistema.

   È possibile utilizzare la risorsa env per definire le variabili di ambiente. In questo caso, la ricetta consente di eseguire con facilità gli script Python dalla riga di comando aggiungendo `c:\python34` al percorso.

   - Il nome della risorsa specifica il nome della variabile di ambiente, PATH in questo esempio.
   - L'attributo `value` specifica il valore della variabile, `c:\\python34` per questo esempio (è necessario utilizzare il carattere di escape \).
   - L'azione `:modify` antepone il valore specificato al valore corrente della variabile.
   - L'attributo `delim` specifica un delimitatore che separa il nuovo valore dal valore esistente, che per questo esempio è `;`.

6. Creare un archivio `.zip` di `installpython`, caricare l'archivio su un bucket S3 e renderlo pubblico. Registrare l'URL dell'archivio per utilizzarlo in seguito. Per ulteriori informazioni, consulta Repository dei libri di ricette.

   I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Crea uno stack per questo esempio nel modo seguente. Puoi anche utilizzare uno stack Windows esistente. È sufficiente aggiornare i libri di ricette, come descritto di seguito.

Creare uno stack

1. Aprire la console AWS OpsWorks Stacks e scegliere Add Stack (Aggiungi stack). Specificare le impostazioni seguenti, accettare i valori predefiniti di altre impostazioni e scegliere Add Stack (Aggiungi stack).

   - Nome — InstallPython
   - Regione: Stati Uniti occidentali (Oregon)

     Questo esempio funzionerà in qualsiasi regione, ma consigliamo di utilizzare Stati Uniti occidentali (Oregon) per i tutorial.
   - Sistema operativo predefinito: Microsoft Windows Server 2012 R2

2. Scegliere Add a layer (Aggiungi un livello) e aggiungere un livello personalizzato allo stack con le impostazioni seguenti.

   - Nome — Python
   - Nome abbreviato: python

3. Aggiungere un'istanza 24/7 con impostazioni predefinite al livello Python e avviarla.

Una volta che l'istanza è online, puoi installare il libro di ricette ed eseguire la ricetta

Per installare il libro di ricette ed eseguire la ricetta

1. Modificare lo stack per abilitare i libri di ricette personalizzati e specificare le impostazioni seguenti.

   - Tipo di repository: S3 Archive.
   - URL del repository: l'URL di archivio del libro di cucina che hai registrato in precedenza.

   Accettare i valori predefiniti per altre impostazioni e scegliere Save (Salva) per aggiornare la configurazione dello stack.

2. Eseguire il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati), che installa la versione più recente dei libri di ricette personalizzati nelle istanze online dello stack. Se è presente una versione precedente del libro di ricette, verrà sovrascritta da questo comando.
3. Eseguire la ricetta utilizzando il comando dello stack Execute Recipes (Esegui ricette) con Recipes to execute (Ricette da eseguire) impostato su **`installpython::default`**. Questo comando avvia un'esecuzione di Chef, con un elenco di esecuzione costituito da `installpython::default`.

   > **Note**
   >
   > Questo esempio utilizza Execute Recipes per comodità, ma in genere AWS OpsWorks Stacks esegue le ricette automaticamente assegnandole all'evento del ciclo di vita appropriato. Puoi eseguire tali ricette attivando manualmente l'evento e Puoi utilizzare un comando dello stack per attivare gli eventi Setup e Configure e un comando di distribuzione per attivare gli eventi Deploy e Undeploy.

4. Per verificare l'installazione, utilizzare RDP per connettersi all'istanza e aprire Esplora risorse.

- Il file system deve avere una directory `C:\Python34`.
- Se si esegue `path` dalla riga di comando, si dovrebbe visualizzare quanto riportato di seguito: `PATH=c:\python34;C:\Windows\system32;...`
- Se si esegue `python --version` dalla riga di comando, dovrebbe restituire `Python 3.4.3`.

Sostituzione degli attributi integrati

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo argomento si applica solo agli stack Linux. Gli attributi integrati sugli stack Windows non possono essere sostituiti.

AWS OpsWorks Stacks installa un set di libri di cucina integrati su ogni istanza. Molti dei libri di ricette integrati supportano i livelli integrati mentre i loro file di attributi definiscono varie impostazioni predefinite di sistema e delle applicazioni, come le impostazioni di configurazione del server Apache. Inserendo queste impostazioni nei file di attributi, puoi personalizzare molte impostazioni di configurazione sostituendo l'attributo integrato corrispondente in uno dei seguenti modi:

- Definire l'attributo in un JSON personalizzato.

  Questo approccio ha il vantaggio di essere semplice e flessibile. Tuttavia, devi immettere manualmente il JSON personalizzato, pertanto non vi sono metodi affidabili per gestire le definizioni degli attributi.
- Implementa un libro di ricette personalizzato e definisci l'attributo in un file di attributi `customize.rb`.

Questo approccio è meno flessibile rispetto all'utilizzo di un JSON personalizzato, ma è più affidabile poiché puoi mettere i libri di ricette personalizzati sotto il controllo del codice sorgente.

In questo argomento viene descritto come utilizzare un file degli attributi del libro di ricette personalizzato per sostituire gli attributi integrati, utilizzando ad esempio il server Apache. Per ulteriori informazioni su come sostituire gli attributi con un JSON personalizzato, consulta Utilizzo di un JSON personalizzato. Per una discussione generale su come sostituire gli attributi, consulta Sostituzione degli attributi.

> **Note**
>
> La sostituzione degli attributi rappresenta il modo migliore per personalizzare le impostazioni di configurazione, ma le impostazioni non sono sempre rappresentate dagli attributi. In questo caso, spesso puoi personalizzare il file di configurazione sostituendo il modello che le ricette integrate utilizzano per creare il file di configurazione. Per vedere un esempio, consulta Sostituzione dei modelli integrati.

Gli attributi integrati in genere rappresentano valori nei file modello che le ricette di impostazione utilizzano per creare i file di configurazione. Ad esempio, una delle ricette di impostazione `apache2`, `default.rb`, usa il modello `apache2.conf.erb` per creare il file di configurazione principale del server Apache, `httpd.conf` (Amazon Linux) o `apache2.conf` (Ubuntu). Di seguito viene riportato un estratto del file modello:

```
...
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests <%= node[:apache][:keepaliverequests] %>
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout <%= node[:apache][:keepalivetimeout] %>
##
## Server-Pool Size Regulation (MPM specific)
```

```
##

...
```

L'impostazione `KeepAliveTimeout` in questo esempio è il valore dell'attributo `[:apache][:keepalivetimeout]`. Il valore predefinito di questo attributo è definito nel file di attributi `apache2`[`apache.rb` del libro di ricette](), come mostrato nell'estratto seguente:

```
...
# General settings
default[:apache][:listen_ports] = [ '80','443' ]
default[:apache][:contact] = 'ops@example.com'
default[:apache][:log_level] = 'info'
default[:apache][:timeout] = 120
default[:apache][:keepalive] = 'Off'
default[:apache][:keepaliverequests] = 100
default[:apache][:keepalivetimeout] = 3
...
```

> **Note**
>
> Per ulteriori informazioni sugli attributi integrati comunemente utilizzati, consulta Attributi dei libri di ricette predefiniti.

Per supportare la sostituzione degli attributi integrati, tutti i libri di ricette integrati contengono un file degli attributi `customize.rb`, incorporato in tutti i moduli tramite una direttiva `include_attribute`. I file `customize.rb` dei libri di ricette integrati non contengono attributi e non avranno alcun effetto sugli attributi incorporati. Per sostituire gli attributi integrati, devi creare un libro di ricette personalizzato con lo stesso nome del libro di ricette integrato e inserire le definizioni degli attributi personalizzati in un file degli attributi denominato `customize.rb`. Questo file ha la precedenza rispetto alla versione integrata ed è incluso in tutti i moduli correlati. Se definisci gli attributi integrati nel tuo `customize.rb`, questi sostituiscono gli attributi integrati corrispondenti.

Questo esempio illustra come sostituire l'attributo `[:apache][:keepalivetimeout]` integrato per impostare il relativo valore su 5 anziché su 3. Puoi utilizzare un approccio simile per qualsiasi attributo integrato. Tuttavia, fai attenzione agli attributi che sostituisci. Ad esempio, la sostituzione degli attributi nello spazio dei nomi `opsworks` potrebbe causare problemi ad alcune ricette integrate.

> **Important**
>
> Non sostituire gli attributi integrati modificando una copia del file degli attributi integrati. Ad esempio, potresti inserire una copia di `apache.rb` nella cartella `apache2/attributes` del tuo libro di ricette personalizzato e modificare alcune impostazioni. Tuttavia, questo file prevale sulla versione integrata e le ricette integrate utilizzeranno la tua versione di `apache.rb`. Se AWS OpsWorks Stacks successivamente modifica il `apache.rb` file integrato, le ricette non otterranno i nuovi valori a meno che non aggiorni manualmente la versione. Utilizzando`customize.rb`, sovrascrivete solo gli attributi specificati; le ricette integrate continuano a ottenere automaticamente up-to-date valori per ogni attributo che non avete sovrascritto.

Per iniziare, crea un libro di ricette personalizzato.

Per creare il libro di ricette

1. Nella directory `opsworks_cookbooks` creare la directory del libro di ricette denominata `apache2` e accedervi.

   Per ignorare gli attributi integrati, il libro di ricette personalizzato deve avere lo stesso nome del libro di ricette integrato, `apache2` in questo esempio.
2. Nella directory `apache2` creare una directory `attributes`.
3. Aggiungere un file denominato `customize.rb` alla directory `attributes` e utilizzarlo per definire gli attributi del libro di ricette integrato che si desidera ignorare. Per questo esempio il file dovrebbe contenere quanto segue:

   ```
   normal[:apache][:keepalivetimeout] = 5
   ```

   > **Important**
   >
   > Per sostituire un attributo integrato, un attributo personalizzato deve essere di tipo `normal` o versione successiva e avere esattamente lo stesso nome di nodo dell'attributo integrato corrispondente. Il tipo `normal` garantisce che l'attributo personalizzato prevalga sugli attributi integrati, tutti di tipo `default`. Per ulteriori informazioni, consulta Precedenza degli attributi.

4. Crea un `.zip` archivio di `opsworks_cookbooks` named `opsworks_cookbooks.zip` e carica l'archivio in un bucket Amazon Simple Storage Service (Amazon S3). Per semplicità, rendere pubblico il file. Registrare l'URL per utilizzarlo successivamente. Puoi anche archiviare i tuoi libri di cucina in un archivio privato di Amazon S3 o in altri tipi di repository. Per ulteriori informazioni, consulta Repository dei libri di ricette.

   I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Per utilizzare l'attributo personalizzato, creare uno stack e installare il libro di ricette.

Per utilizzare l'attributo personalizzato

1. Aprire la console AWS OpsWorks Stacks e scegliere Add Stack (Aggiungi stack).
2. Specificare le seguenti impostazioni standard.

   - Nome: ApacheConfig
   - Regione: Stati Uniti occidentali (Oregon)

     Puoi inserire il tuo stack in qualsiasi regione, ma per i tutorial consigliamo US West (Oregon).
   - Chiave SSH predefinita: una coppia di EC2 chiavi

     Se devi creare una coppia di EC2 chiavi, consulta Amazon EC2 Key Pairs. La coppia di chiavi deve appartenere alla stessa regione AWS dello stack.

   Scegliere Advanced (Avanzate) **>>**, impostare Use custom Chef cookbooks (Usa libri di ricette Chef personalizzati) su Yes (Sì), quindi specificare le seguenti impostazioni.

   - Tipo di repository: Http Archive
   - URL del repository: l'URL dell'archivio del libro di cucina che hai registrato in precedenza

   Accettare i valori predefiniti di altre impostazioni e scegliere Add Stack (Aggiungi stack) per creare lo stack.

> **Note**
>
> Questo esempio utilizza il sistema operativo predefinito, ovvero Amazon Linux. Se preferisci, puoi utilizzare Ubuntu. L'unica differenza è che nei sistemi Ubuntu, la ricetta Setup integrata produce un file di configurazione con le stesse impostazioni denominato `apache2.conf` e lo inserisce nella directory `/etc/apache2`.

3. Scegliete Aggiungi un layer, quindi aggiungete un layer Java App Server con le impostazioni predefinite allo stack.
4. Aggiungere un'istanza 24/7 con impostazioni predefinite al livello, quindi avviare l'istanza.

   Un'istanza t2.micro è sufficiente per questo esempio.
5. Una volta che l'istanza è online, connettersi a essa con SSH. Il file `httpd.conf` è nella directory `/etc/httpd/conf`. Se si esamina il file, dovrebbe essere visibile l'impostazione `KeepAliveTimeout` personalizzata. Le restanti impostazioni utilizzeranno i valori predefiniti presenti nel file `apache.rb` integrato. La parte pertinente di `httpd.conf` dovrebbe risultare simile alla seguente:

```
...
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5
...
```

Sostituzione dei modelli integrati

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo argomento si applica solo agli stack Linux. I modelli integrati sugli stack Windows non possono essere sostituiti.

Le ricette integrate di AWS OpsWorks Stacks utilizzano modelli per creare file su istanze, principalmente file di configurazione per server, come Apache. Ad esempio, le ricette `apache2` utilizzano il modello `apache2.conf.erb` per creare il file di configurazione primario del server Apache, `httpd.conf` (Amazon Linux) o `apache2.conf` (Ubuntu).

La maggior parte delle impostazioni di configurazione in questi modelli è rappresentata da attributi, pertanto il modo migliore per personalizzare un file di configurazione consiste nel sostituire gli attributi integrati appropriati. Per vedere un esempio, consulta Sostituzione degli attributi integrati. Tuttavia, se le impostazioni che desideri personalizzare non sono rappresentate da attributi integrati o non sono presenti nel modello, devi sostituire il modello stesso. In questo argomento viene descritto come sostituire un modello integrato per specificare un'impostazione di configurazione Apache personalizzata.

Puoi fornire risposte di errore personalizzate ad Apache aggiungendo impostazioni `ErrorDocument` al file `httpd.conf`. `apache2.conf.erb` contiene solo alcuni esempi commentati, come illustrato di seguito:

```
...
#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
...
```

Poiché queste impostazioni sono commenti hardcoded, non puoi specificare valori personalizzati sostituendo attributi, ma devi sostituire il modello stesso. Tuttavia, diversamente da quanto avviene con gli attributi, non vi è alcun modo per sostituire parti specifiche di un file di modello. Devi creare

un libro di ricette personalizzato con lo stesso nome della versione integrata, copiare il file di modello nella stessa sottodirectory e modificare il file in base alle esigenze. Questo argomento mostra come sostituire `apache2.conf.erb` per fornire una risposta personalizzata all'errore 500. Per una discussione generale sulla sostituzione dei modelli, consulta Utilizzo di modelli personalizzati .

> **Important**
>
> Quando sostituisci un modello integrato, le ricette integrate utilizzano la tua versione personalizzata del modello anziché la versione integrata. Se AWS OpsWorks Stacks aggiorna il modello integrato, il modello personalizzato non è sincronizzato e potrebbe non funzionare correttamente. AWS OpsWorks Stacks non apporta tali modifiche spesso e, quando un modello cambia, AWS OpsWorks Stacks elenca le modifiche e ti offre la possibilità di eseguire l'aggiornamento a una nuova versione. Ti consigliamo di monitorare il repository AWS OpsWorks Stacks per scoprire le eventuali modifiche e aggiornare manualmente il modello personalizzato in base alle esigenze. Il repository include un ramo distinto per ciascuna versione Chef supportata, pertanto assicurati di essere nel ramo corretto.

Per iniziare, crea un libro di ricette personalizzato.

Per creare il libro di ricette

1. Nella directory `opsworks_cookbooks` creare la directory del libro di ricette denominata `apache2` e accedervi. Per ignorare i modelli integrati, il libro di ricette personalizzato deve avere lo stesso nome del libro di ricette integrato, `apache2` in questo esempio.

   > **Note**
   >
   > Se hai già completato la procedura guidata Sostituzione degli attributi integrati, puoi utilizzare lo stesso libro di ricette `apache2` per questo esempio e saltare il Passaggio 2.

2. Creare un file `metadata.rb` con il seguente contenuto e salvarlo nella directory `apache2`.

   ```
   name "apache2"
   version "0.1.0"
   ```

3. Nella directory `apache2` creare una directory `templates/default`.

> **Note**
>
> La `templates/default` directory funziona per le istanze Amazon Linux, che utilizzano il `apache2.conf.erb` modello predefinito. Le istanze Ubuntu 14.04 utilizzano un modello `apache2.conf.erb` specifico del sistema operativo, che si trova nella directory `templates/ubuntu-14.04`. Se si desidera che la personalizzazione venga applicata alle istanze Ubuntu 14.04, inoltre, è necessario sovrascrivere anche tale modello.

4. Copiare il modello `apache2.conf.erb` integrato nella directory `templates/default`. Aprire il file modello, rimuovere i commenti della riga `ErrorDocument 500` e fornire un messaggio di errore personalizzato, come segue:

```
...
ErrorDocument 500 "A custom error message."
#ErrorDocument 404 /missing.html
...
```

5. Crea un `.zip` archivio di `opsworks_cookbooks` named`opsworks_cookbooks.zip`, quindi carica il file in un bucket Amazon Simple Storage Service (Amazon S3). Per semplicità, rendere pubblico l'archivio. Registrare l'URL dell'archivio per utilizzarlo in seguito. Puoi anche archiviare i tuoi libri di cucina in un archivio privato di Amazon S3 o in altri tipi di repository. Per ulteriori informazioni, consulta Repository dei libri di ricette.

   I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

> **Note**
>
> Per semplicità, questo esempio aggiunge un messaggio di errore hardcoded al modello. Per modificarlo, devi modificare il modello e reinstallare il libro di ricette. Per usufruire di una maggiore flessibilità, puoi definire un attributo personalizzato predefinito per la stringa di errore nel file degli attributi `customize.rb` del libro di ricette personalizzato e assegnare il valore di tale attributo a `ErrorDocument 500`. Ad esempio, se assegni un nome all'attributo `[:apache][:custom][:error500]`, la riga corrispondente in `apache2.conf.erb` sarà simile alla seguente:

```
...
ErrorDocument 500 <%= node[:apache][:custom][:error500] %>
#ErrorDocument 404 /missing.html
...
```

Quindi puoi modificare il messaggio di errore personalizzato in qualsiasi momento sostituendo `[:apache][:custom][:error500]`. Se utilizzi il JSON personalizzato per sostituire l'attributo, non devi neanche toccare il libro di ricette.

Per utilizzare il modello personalizzato, creare uno stack e installare il libro di ricette.

Per utilizzare il modello personalizzato

1. Aprire la console AWS OpsWorks Stacks e scegliere Add Stack (Aggiungi stack).
2. Specificare le seguenti impostazioni standard:

   - Nome: ApacheTemplate
   - Regione: Stati Uniti occidentali (Oregon)
   - Chiave SSH predefinita: una coppia di chiavi Amazon Elastic Compute Cloud EC2 (Amazon)

     Se devi creare una coppia di EC2 chiavi Amazon, consulta Amazon EC2 Key Pairs. La coppia di chiavi deve appartenere alla stessa regione AWS dell'istanza.

   Scegliere Advanced (Avanzate) **>>**, scegliere Use custom Chef cookbooks (Utilizza libri di ricette Chef personalizzati) per specificare le impostazioni seguenti:

   - Tipo di repository: Http Archive
   - URL del repository: l'URL dell'archivio del libro di cucina che hai registrato in precedenza

   Accettare i valori predefiniti di altre impostazioni e scegliere Add Stack (Aggiungi stack) per creare lo stack.
3. Scegliete Aggiungi un layer, quindi aggiungete un layer Java App Server allo stack con le impostazioni predefinite.
4. Aggiungere un'istanza 24/7 con impostazioni predefinite al livello, quindi avviare l'istanza.

Un'istanza t2.micro è sufficiente per questo esempio.

5. Una volta che l'istanza è online, connettersi a essa con SSH. Il file `httpd.conf` è nella directory `/etc/httpd/conf`. Il file deve contenere la tua impostazione `ErrorDocument` personalizzata, che avrà un aspetto simile al seguente:

```
...
# Some examples:
ErrorDocument 500 "A custom error message."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
...
```

Bilanciamento del carico di un livello

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks offre due opzioni di bilanciamento del carico, Elastic Load Balancing HAProxye, che vengono in genere utilizzate per bilanciare il carico tra le istanze di un livello di application server. Questo argomento descrive i vantaggi e le limitazioni di ciascuna opzione, in modo da permetterti di scegliere l'opzione più adatta quando devi aggiungere il bilanciamento del carico a un livello. In alcuni casi, è consigliabile utilizzare entrambe le opzioni.

Terminazione SSL

Il HAProxy livello integrato non gestisce la terminazione SSL; è necessario terminare SSL sui server. Il vantaggio di questo approccio è che il traffico rimane crittografato finché non raggiunge i server. Tuttavia, i server devono gestire la decrittografia e ciò aumenta il carico dei server. Inoltre, devi inserire i certificati SSL nei server di applicazioni, che risultano più accessibili agli utenti.

Con Elastic Load Balancing, puoi terminare il protocollo SSL sul sistema di bilanciamento del carico. Ciò riduce il carico sui server delle applicazioni, ma il traffico tra il load balancer e il server non è crittografato. Elastic Load Balancing consente anche di terminare il protocollo SSL sul server, ma è piuttosto complicato da configurare.

Dimensionamento

Se il traffico in entrata supera la capacità di un sistema di bilanciamento del HAProxy carico, è necessario aumentarne la capacità manualmente.

Elastic Load Balancing si ridimensiona automaticamente per gestire il traffico in entrata. Per garantire che un sistema di bilanciamento del carico Elastic Load Balancing abbia una capacità sufficiente per gestire il carico previsto quando viene messo online per la prima volta, puoi preriscaldarlo.

Errore del sistema di bilanciamento del carico

Se l'istanza che ospita il HAProxy server si guasta, l'intero sito potrebbe essere offline fino al riavvio dell'istanza.

Elastic Load Balancing è più resistente ai guasti di. HAProxy Ad esempio, fornisce nodi di bilanciamento del carico in ogni zona di disponibilità con istanze registrate EC2 . Se il servizio in una regione in una zona viene interrotto, gli altri nodi continuano a gestire il traffico in entrata. Per ulteriori informazioni, consulta Elastic Load Balancing Concepts.

Tempo di inattività

Entrambi i sistemi di bilanciamento del carico terminano una connessione se un server è inattivo per un periodo maggiore del valore del timeout di inattività specificato.

- HAProxy — Il valore del timeout di inattività non ha un limite massimo.
- Elastic Load Balancing: il valore di timeout di inattività predefinito è 60 secondi, con un massimo di 3600 secondi (60 minuti).

Il limite di inattività di Elastic Load Balancing è sufficiente per la maggior parte degli scopi. Ti consigliamo di utilizzarlo HAProxy se hai bisogno di un timeout di inattività più lungo. Per esempio:

- Una connessione HTTP di lunga durata utilizzata per le notifiche push.
- Un'interfaccia di amministrazione che puoi utilizzare per eseguire attività che potrebbero richiedere più di 60 minuti.

Mappatura basata su URL

Potresti aver bisogno di utilizzare un sistema di bilanciamento del carico che inoltri una richiesta in entrata a un determinato server in base all'URL della richiesta. Ad esempio, supponiamo che tu abbia un gruppo di dieci server di applicazioni che supportano un'applicazione di commercio online. Otto server gestiscono il catalogo e due server gestiscono i pagamenti. Vuoi indirizzare tutte le richieste HTTP relative ai pagamenti ai server dei pagamenti, in base all'URL della richiesta. In questo caso, dovresti indirizzare tutto ciò URLs che include «pagamento» o «checkout» a uno dei server di pagamento.

Con HAProxy, puoi utilizzare la mappatura basata su URL per indirizzare il URLs contenuto di una stringa specificata a server particolari. Per utilizzare la mappatura basata su URL con AWS OpsWorks Stacks, devi creare un file di HAProxy configurazione personalizzato sovrascrivendo il modello nel `haproxy-default.erb` ricettario integrato. `haproxy` Per ulteriori informazioni, consulta il Manuale di configurazione e. HAProxy Utilizzo di modelli personalizzati Non puoi utilizzare la mappatura basata su URL per le richieste HTTPS. Una richiesta HTTPS è crittografata, quindi non HAProxy ha modo di esaminare l'URL della richiesta.

Elastic Load Balancing ha un supporto limitato per la mappatura degli URL. Per ulteriori informazioni, consulta Configurazioni del listener per Elastic Load Balancing.

Raccomandazione: consigliamo di utilizzare Elastic Load Balancing per il bilanciamento del carico a meno che non si abbiano requisiti che possono essere gestiti solo da. HAProxy In tal caso, l'approccio migliore potrebbe essere quello di combinare le due cose utilizzando Elastic Load Balancing come sistema di bilanciamento del carico front-end che distribuisce il traffico in entrata a un set di server. HAProxy Per farlo:

- Configura un' HAProxy istanza in ciascuna delle zone di disponibilità dello stack per distribuire le richieste ai server delle applicazioni della zona.
- Assegna le HAProxy istanze a un sistema di bilanciamento del carico Elastic Load Balancing, che quindi distribuisce le richieste in entrata ai sistemi di bilanciamento del carico. HAProxy

Questo approccio consente di utilizzare la mappatura basata su URL per distribuire diversi tipi HAProxy di richieste ai server applicativi appropriati. Tuttavia, se uno dei HAProxy server va offline, il sito continuerà a funzionare perché il load balancer Elastic Load Balancing distribuisce automaticamente il traffico in entrata ai server integri. HAProxy Tieni presente che devi utilizzare

Elastic Load Balancing come load balancer front-end; un HAProxy server non può distribuire richieste ad altri server. HAProxy

Migrazione da Chef Server a Stacks AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Poiché AWS OpsWorks Stacks è basato su Chef, la migrazione da Chef Server a AWS OpsWorks Stacks è relativamente semplice. Questo argomento presenta le linee guida per modificare il codice di Chef Server in modo da utilizzarlo con AWS OpsWorks Stacks.

> **Note**
>
> Sconsigliamo di eseguire la migrazione agli stack tramite versioni di Chef precedenti alla 11.10, che sono basate su chef-solo e non supportano la ricerca o i contenitori di dati.

Argomenti

- Mappatura di ruoli a livelli
- Utilizzo di contenitori di dati
- Utilizzo della ricerca di Chef
- Gestione di libri di ricette e ricette
- Utilizzo di ambienti Chef

Mappatura di ruoli a livelli

Chef Server utilizza ruoli per rappresentare e gestire istanze con lo stesso scopo e la stessa configurazione, ad esempio un set di istanze ciascuna delle quali ospita un server applicazioni Java. Un livello AWS OpsWorks Stacks ha essenzialmente lo stesso scopo di un ruolo Chef. Un livello è

un modello per creare un set di istanze Amazon Elastic Compute Cloud EC2 (Amazon) con la stessa configurazione, pacchetti installati, procedura di distribuzione delle applicazioni e così via.

AWS OpsWorks Stacks include un set di livelli integrati per diversi tipi di server applicativi, un HAProxy load balancer, un database master MySQL e un master di monitoraggio Ganglia. Ad esempio, il livello Java App Server integrato è un modello per la creazione di istanze che ospitano un server Tomcat.

Per migrare a AWS OpsWorks Stacks, è necessario associare ogni ruolo a un livello che fornisca funzionalità equivalenti. Per alcuni ruoli, potresti essere in grado di utilizzare semplicemente uno dei livelli predefiniti. Altri ruoli potrebbero richiedere diversi gradi personalizzazione. Inizia esaminando le funzionalità dei livelli predefiniti, tra cui le ricette associate a ognuno, per determinare se uno di questi livelli sia in grado di offrire almeno parte delle funzionalità del tuo ruolo. Per ulteriori informazioni sui livelli predefiniti, consulta Livelli e AWS OpsWorks Stacks Layer Reference. Per esaminare le ricette integrate, consulta l'archivio pubblico di AWS OpsWorks Stacks. GitHub

Il modo di procedere dipende dal grado di corrispondenza di un livello a ogni ruolo, come descritto di seguito.

Un livello predefinito supporta tutte le funzionalità del ruolo

Se necessario, puoi utilizzare direttamente il livello predefinito, con personalizzazioni minime. Ad esempio, se un ruolo supporta un server Tomcat, le ricette del livello Java App Server potrebbero già gestire tutte le attività del ruolo, magari con qualche modesta personalizzazione. Ad esempio, puoi fare in modo che le ricette predefinite del livello utilizzino impostazioni di configurazione di Tomcat o Apache personalizzate sostituendo gli attributi o i modelli appropriati.

Un livello predefinito supporta alcune, ma non tutte, le funzionalità del ruolo

Potresti essere in grado di utilizzare un livello predefinito estendendolo. Questa situazione comporta in genere l'implementazione di ricette personalizzate per supportare le funzionalità mancanti e assegnare le ricette agli eventi del ciclo di vita del livello. Ad esempio, supponiamo che il tuo ruolo installi un server Redis nelle stesse istanze che ospitano un server Tomcat. È possibile estendere il livello Java App Server in modo che corrisponda alla funzionalità del ruolo implementando una ricetta personalizzata per installare Redis sulle istanze del livello e assegnando la ricetta all'evento Setup del livello.

Nessun livello predefinito supporta in modo adeguato le funzionalità del ruolo

Implementa un livello personalizzato. Ad esempio, supponiamo che il tuo ruolo supporti un server di database MongoDB, che non è supportato da alcuno dei livelli predefiniti. Puoi fornire questo

supporto implementando ricette per l'installazione dei pacchetti necessari, la configurazione del server e così via e quindi assegnare le ricette a eventi del ciclo di vita di un livello personalizzato. In genere, puoi utilizzare almeno una delle ricette del ruolo a questo scopo. Per ulteriori informazioni su come implementare un livello personalizzato, consulta Creazione di un livello server Tomcat personalizzato.

Utilizzo di contenitori di dati

Chef Server ti permette di passare dati definiti dall'utente alle ricette utilizzando contenitori di dati.

- I dati vengono archiviati con i libri di ricette e Chef li installa in ogni istanza.
- Puoi utilizzare contenitori di dati crittografati per dati sensibili, ad esempio le password.

AWS OpsWorks Stacks supporta i data bag; le ricette possono recuperare i dati utilizzando esattamente lo stesso codice di Chef Server. Tuttavia, il supporto presenta le limitazioni e differenze seguenti:

- I contenitori di dati sono supportati solo su stack Linux Chef 11.10 e versioni successive.

  Gli stack Windows e Linux che eseguono versioni precedenti di Chef non supportano i contenitori di dati.
- Non puoi archiviare contenitori di dati nel repository dei libri di ricette.

  Devi invece utilizzare codice JSON personalizzato per gestire i dati del contenitore di dati.
- AWS OpsWorks Stacks non supporta sacchetti di dati crittografati.

  Se devi archiviare dati sensibili in forma crittografata, ad esempio password o certificati, ti consigliamo di archiviarli in un bucket S3 privato. Potrai quindi creare una ricetta personalizzata che utilizza l'SDK Amazon per Ruby per recuperare i dati. Per vedere un esempio, consulta Utilizzo dell'SDK for Ruby.

Per ulteriori informazioni, consulta Utilizzo di contenitori di dati.

Utilizzo della ricerca di Chef

Chef Server archivia le informazioni di configurazione degli stack, come indirizzi IP e configurazioni dei ruoli, nel server. Le ricette utilizzano Chef search per recuperare questi dati. AWS OpsWorks Stacks utilizza un approccio leggermente diverso. Ad esempio, gli stack Linux Chef 11.10 sono basati

sulla modalità locale client di Chef, un'opzione client di Chef che esegue una versione leggera di Chef Server, spesso denominata Chef Zero, in locale nell'istanza. Chef Zero supporta la ricerca sui dati archiviati nell'oggetto nodo dell'istanza.

Invece di archiviare i dati dello stack su un server remoto, AWS OpsWorks Stacks aggiunge un set di attributi di configurazione e distribuzione dello stack all'oggetto nodo di ciascuna istanza per ogni evento del ciclo di vita. Questi attributi rappresentano uno snapshot della configurazione dello stack. Gli attributi utilizzano la stessa sintassi di Chef Server e rappresentano la maggior parte dei dati che le ricette devono recuperare dal server.

Spesso non è necessario modificare il codice dipendente dalla ricerca delle ricette per Stacks. AWS OpsWorks Poiché Chef search opera sull'oggetto node, che include la configurazione dello stack e gli attributi di distribuzione, le query di ricerca in AWS OpsWorks Stacks di solito funzionano esattamente come con Chef Server.

L'eccezione principale è causata dal fatto che gli attributi di configurazione e distribuzione dello stack contengono solo dati di cui AWS OpsWorks Stacks è a conoscenza quando installa gli attributi sull'istanza. Se crei o modifichi un attributo localmente su una particolare istanza, tali modifiche non vengono propagate agli AWS OpsWorks Stacks e non vengono incorporate negli attributi di configurazione e distribuzione dello stack installati sulle altre istanze. Puoi utilizzare la ricerca per recuperare il valore dell'attributo solo nell'istanza specifica. Per ulteriori informazioni, consulta Utilizzo della ricerca di Chef.

Per compatibilità con Chef Server, AWS OpsWorks Stacks aggiunge un set di `role` attributi all'oggetto node, ognuno dei quali contiene uno degli attributi di livello dello stack. Se la ricetta utilizza `roles` come chiave di ricerca, non dovrai modificare il codice di ricerca. La query restituisce automaticamente i dati per il livello corrispondente. Ad esempio, entrambe le query seguenti restituiscono gli attributi del livello `php-app`.

```
phpserver = search(:node, "layers:php-app").first
```

```
phpserver = search(:node, "roles:php-app").first
```

Gestione di libri di ricette e ricette

AWS OpsWorks Stacks e Chef Server gestiscono i libri di cucina e le ricette in modo leggermente diverso. Con Server Chef:

- Devi fornire tutti i libri di ricette, implementandoli personalmente o utilizzando libri di ricette della community.
- Devi archiviare i libri di ricette nel server.
- Devi eseguire le ricette manualmente o in base a una pianificazione regolare.

Con AWS OpsWorks Stacks:

- AWS OpsWorks Stacks fornisce uno o più libri di cucina per ciascuno dei livelli integrati. Questi libri di ricette gestiscono le attività standard, come l'installazione e la configurazione del software di un livello predefinito e la distribuzione di app.

  Per gestire le attività che non vengono eseguite dai libri di ricette predefiniti, devi aggiungere libri di ricette personalizzati allo stack o utilizzare libri di ricette della community.
- I libri di cucina AWS OpsWorks Stacks vengono archiviati in un archivio remoto, ad esempio un bucket S3 o un repository Git.

  Per ulteriori informazioni, consulta Archiviazione di libri di ricette.
- È possibile eseguire le ricette manualmente, ma in genere AWS OpsWorks Stacks esegue le ricette automaticamente in risposta a una serie di eventi del ciclo di vita che si verificano in punti chiave durante il ciclo di vita di un'istanza.

  Per ulteriori informazioni, consulta Esecuzione di ricette.
- AWS OpsWorks Stacks supporta Berkshelf solo sugli stack Chef 11.10. Se utilizzi Berkshelf per gestire le dipendenze dei libri di ricette, non puoi utilizzare stack che eseguono Chef 11.4 o versioni precedenti.

  Per ulteriori informazioni, consulta Utilizzo di Berkshelf.

Argomenti

- Archiviazione di libri di ricette
- Esecuzione di ricette

Archiviazione di libri di ricette

Con Chef Server, devi archiviare i libri di ricette nel server e distribuirli dal server alle istanze. Con AWS OpsWorks Stacks, memorizzi i libri di cucina in un repository: un archivio S3 o HTTP o un

repository Git o Subversion. Quando installi i libri di cucina, specifichi le informazioni di cui AWS OpsWorks Stacks ha bisogno per scaricare il codice dal repository nelle istanze di uno stack.

Per eseguire la migrazione da Chef Server, devi inserire i libri di ricette in uno di questi repository. Per informazioni su come strutturare un repository di libri di ricette, consulta Repository dei libri di ricette.

Esecuzione di ricette

In AWS OpsWorks Stacks, ogni livello ha una serie di eventi del ciclo di vita (Setup, Configure, Deploy, Undeploy e Shutdown), ognuno dei quali si verifica in un momento chiave durante il ciclo di vita di un'istanza. Per eseguire una ricetta personalizzata, in genere devi assegnarla all'evento appropriato nel livello corretto. Quando si verifica l'evento, AWS OpsWorks Stacks esegue le ricette associate. Ad esempio, poiché l'evento Setup (Impostazione) si verifica dopo che un'istanza completa la procedura di avvio, in genere assegnerai ricette a questo evento per l'esecuzione di attività come l'installazione e la configurazione di pacchetti e l'avvio di servizi.

Puoi eseguire manualmente le ricette utilizzando il comando dello stack Execute Recipes (Esegui ricette). Questo comando è utile per lo sviluppo e i test, ma puoi utilizzarlo anche per eseguire ricette che non sono mappate ad alcun evento del ciclo di vita. Puoi utilizzare il comando Execute Recipes (Esegui ricette) anche per attivare manualmente eventi Setup (Impostazione) e Configure (Configurazione).

Oltre alla console AWS OpsWorks Stacks, puoi utilizzare la CLI AWS SDKso eseguire ricette. Questi strumenti supportano tutte le azioni API di AWS OpsWorks Stacks, ma sono più semplici da utilizzare rispetto all'API. Utilizza il comando create-deployment dell'interfaccia a riga di comando per attivare un evento del ciclo di vita, che esegue tutte le ricette associate. Puoi utilizzare questo comando anche per eseguire una o più ricette senza attivare alcun evento. Il codice SDK equivalente dipende dal linguaggio specifico, ma è generalmente simile al comando dell'interfaccia a riga di comando.

Gli esempi seguenti presentano due modi di utilizzare il comando `create-deployment` dell'interfaccia a riga di comando per automatizzare la distribuzione dell'applicazione.

- Distribuzione dell'app in base a una pianificazione regolare aggiungendo un livello personalizzato con una singola istanza allo stack.

  Aggiunta di una ricetta di impostazione personalizzata al livello che crea un processo `cron` nell'istanza per eseguire il comando in base a una pianificazione specificata. Per un esempio di come utilizzare una ricetta per creare un processo `cron`, consulta Esecuzione di processi Cron sulle istanze Linux.

- Aggiunta di un'attività alla pipeline di integrazione continua che utilizza il comando `create-deployment` dell'interfaccia a riga di comando per distribuire l'app.

Utilizzo di ambienti Chef

AWS OpsWorks Stacks non supporta gli ambienti Chef; `node.chef_environment` restituisce sempre. `_default`

AWS OpsWorks Stacks Layer Reference

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ogni istanza distribuita da AWS OpsWorks Stacks deve far parte di almeno un livello, che definisce il ruolo dell'istanza nello stack e controlla i dettagli relativi alla configurazione e alla configurazione dell'istanza, all'installazione dei pacchetti, alla distribuzione delle applicazioni e così via. Per ulteriori informazioni su come utilizzare gli AWS OpsWorks Stack per creare e gestire i livelli, consulta. Livelli

La descrizione di ogni livello include un elenco delle ricette integrate che AWS OpsWorks Stacks esegue per ciascuno degli eventi del ciclo di vita del livello. Queste ricette sono memorizzate in opsworks-cookbooks. https://github.com/aws/ Nota che gli elenchi includono solo quelle ricette che vengono eseguite direttamente da Stacks. AWS OpsWorks le quali talvolta eseguono ricette dipendenti, che non sono elencate. Per visualizzare l'elenco completo delle ricette per un determinato evento, tra cui ricette dipendenti e personalizzate, esamina l'elenco di esecuzione nell'appropriato log Chef dell'evento del ciclo di vita.

Argomenti

- HAProxy Riferimento al livello
- HAProxy AWS OpsWorks Stacks Layer
- Riferimento ai livelli MySQL
- Livello MySQL OpsWorks

- Documentazione di riferimento sul livello del server di applicazioni
- Livelli del server di applicazioni
- Riferimento al livello del cluster ECS
- Documentazione di riferimento sui livelli personalizzati
- Documentazione di riferimento su altri livelli
- Altri livelli

HAProxy Riferimento al livello

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Un HAProxy livello utilizza HAProxy, un sistema di bilanciamento del TCP/HTTP carico affidabile e ad alte prestazioni, per fornire servizi proxy e di bilanciamento del carico ad alta disponibilità per applicazioni basate su TCP e HTTP. È particolarmente utile per quei siti web che devono eseguire il crawling con carichi di lavoro molto elevati e che richiedono persistenza o elaborazione di livello 7.

HAProxy monitora il traffico e visualizza le statistiche e lo stato delle istanze associate su una pagina Web. Per impostazione predefinita, l'URI è http://*DNSName*/haproxy? stats, *DNSName* dov'è il nome DNS dell' HAProxy istanza.

Short name (Nome breve): lb

Compatibilità: un HAProxy livello è compatibile con i seguenti livelli: custom, db-master e memcached.

Porte aperte: HAProxy consente l'accesso pubblico alle porte 22 (SSH), 80 (HTTP) e 443 (HTTPS).

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): attivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): no

Gruppo di sicurezza predefinito: AWS-OpsWorks-LB-Server

Configurazione: per configurare un HAProxy livello, è necessario specificare quanto segue:

- URI Health check (impostazione predefinita: http://*DNSName*/).
- URI delle statistiche (impostazione predefinita: http://*DNSName*/haproxy? statistiche).
- Password statistiche (facoltativa).
- Metodo controllo dello stato (facoltativo). Per impostazione predefinita, HAProxy utilizza il metodo HTTP OPTIONS. Puoi anche specificare GET o HEAD.
- Abilita statistiche (facoltativo)
- Porte. Per impostazione predefinita, AWS OpsWorks Stacks è configurato HAProxy per gestire sia il traffico HTTP che HTTPS. Puoi configurare HAProxy per gestire solo l'uno o l'altro sovrascrivendo il modello di configurazione Chef,. `haproxy.cfg.erb`

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- haproxy

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version

- haproxy::configure

Deploy recipes (Ricette di distribuzione):

- deploy::default
- haproxy::configure

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default
- haproxy::stop

Installation (Installazione):

- AWS OpsWorks Stacks utilizza il programma di installazione del pacchetto dell'istanza per l'installazione nelle posizioni HAProxy predefinite.
- Per indirizzare i file di log in un percorso specificato, è necessario configurare syslog. Per ulteriori informazioni, consulta HAProxy.

HAProxy AWS OpsWorks Stacks Layer

> **Note**
>
> Questo livello è disponibile solo per stack Chef 11 e stack basati su Linux di versioni precedenti.

Lo AWS OpsWorks Stacks HAProxy layer è un layer AWS OpsWorks Stacks che fornisce un modello per le istanze che ospitano un server: un bilanciamento del carico affidabile e ad alte prestazioni. HAProxy TCP/HTTP Un'istanza small è in genere sufficiente per gestire tutto il traffico del server di applicazioni.

> **Note**
>
> Gli stack sono limitati a una singola regione. Per distribuire l'applicazione tra più regioni, devi creare uno stack distinto per ogni regione.

Per creare un livello HAProxy

1. Nel riquadro di navigazione, fare clic su Layers (Livelli).
2. Nella pagina Layers (Livelli), fare clic su Add a Layer (Aggiungi un livello) o + Layer (+ Livello). Per Tipo di livello, selezionate HAProxy.

Il livello è associato alle impostazioni di configurazione opzionali seguenti.

HAProxy statistiche

Specifica se il livello raccoglie e visualizza le statistiche. Il valore predefinito è Yes (Sì).

Statistics URL (URL statistiche)

Percorso dell'URL della pagina delle statistiche. L'URL completo è http://*DNSNameStatisticsPath*, dove *DNSName* è il nome DNS dell'istanza associata. Il *StatisticsPath* valore predefinito è /haproxy? stats, che corrisponde a qualcosa come: http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/haproxy?stats.

Statistics user name (Nome utente statistiche)

Il nome utente della pagina delle statistiche, che devi fornire per visualizzare la pagina delle statistiche. Il valore predefinito è «opsworks».

Statistics password (Password statistiche)

Password della pagina relativa alle statistiche, che deve essere specificata per poter visualizzare la pagina delle statistiche. Il valore predefinito è una stringa generata casualmente.

Health check URL (URL controllo stato)

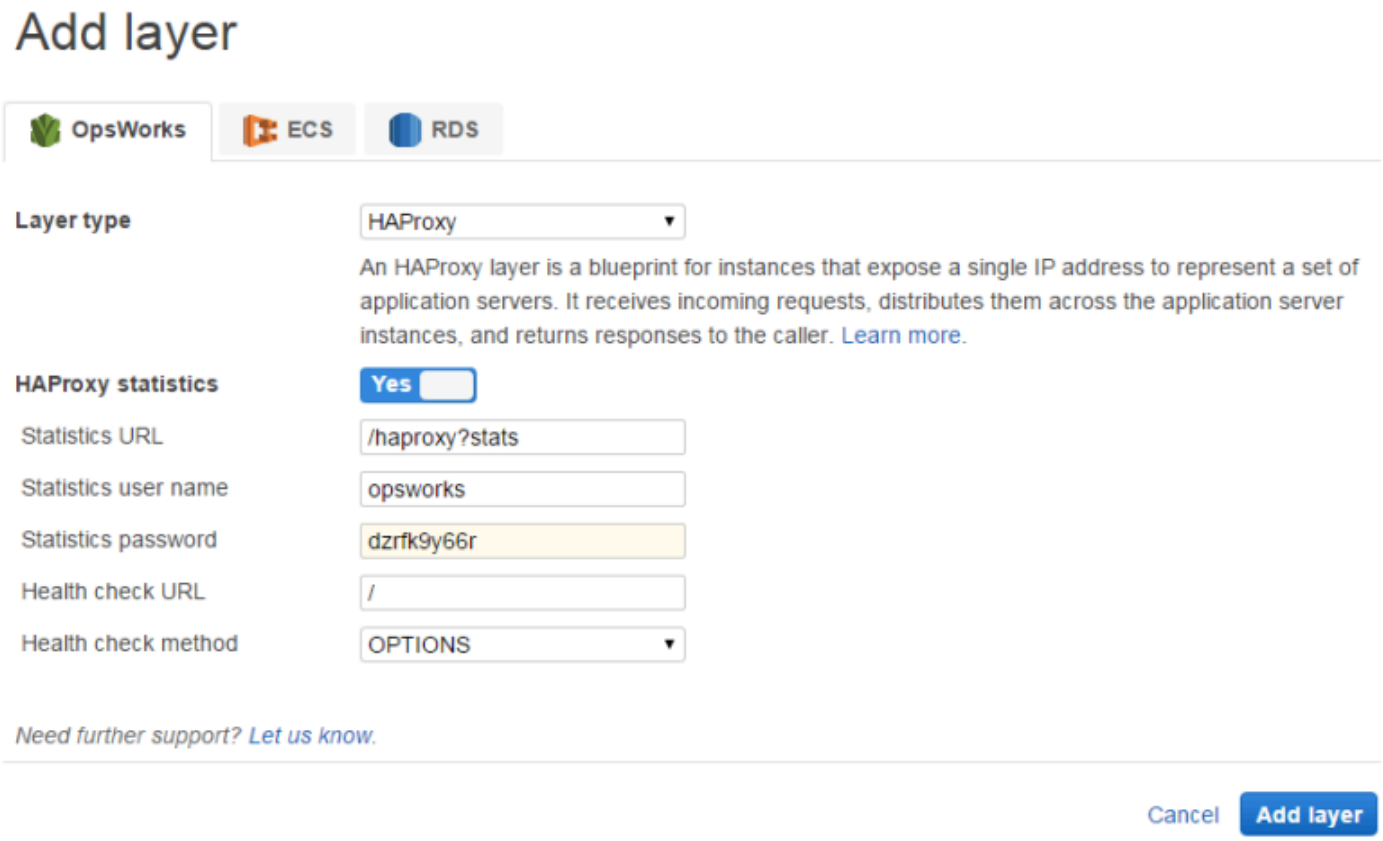
Il suffisso URL del controllo dello stato di salute. HAProxy utilizza questo URL per chiamare periodicamente un metodo HTTP su ogni istanza dell'Application Server per determinare se l'istanza funziona. Se il controllo di integrità fallisce, HAProxy interrompe l'indirizzamento del traffico verso l'istanza fino al riavvio, manualmente o tramite riparazione automatica. Il valore predefinito per il suffisso URL è «/», che corrisponde alla home page dell'istanza del server: http:///. *DNSName*

Health check method (Metodo controllo stato)

Metodo HTTP da utilizzare per controllare se le istanze funzionano. Il valore predefinito è OPTIONS, ma è anche possibile specificare GET o HEAD. Per ulteriori informazioni, consulta httpchk.

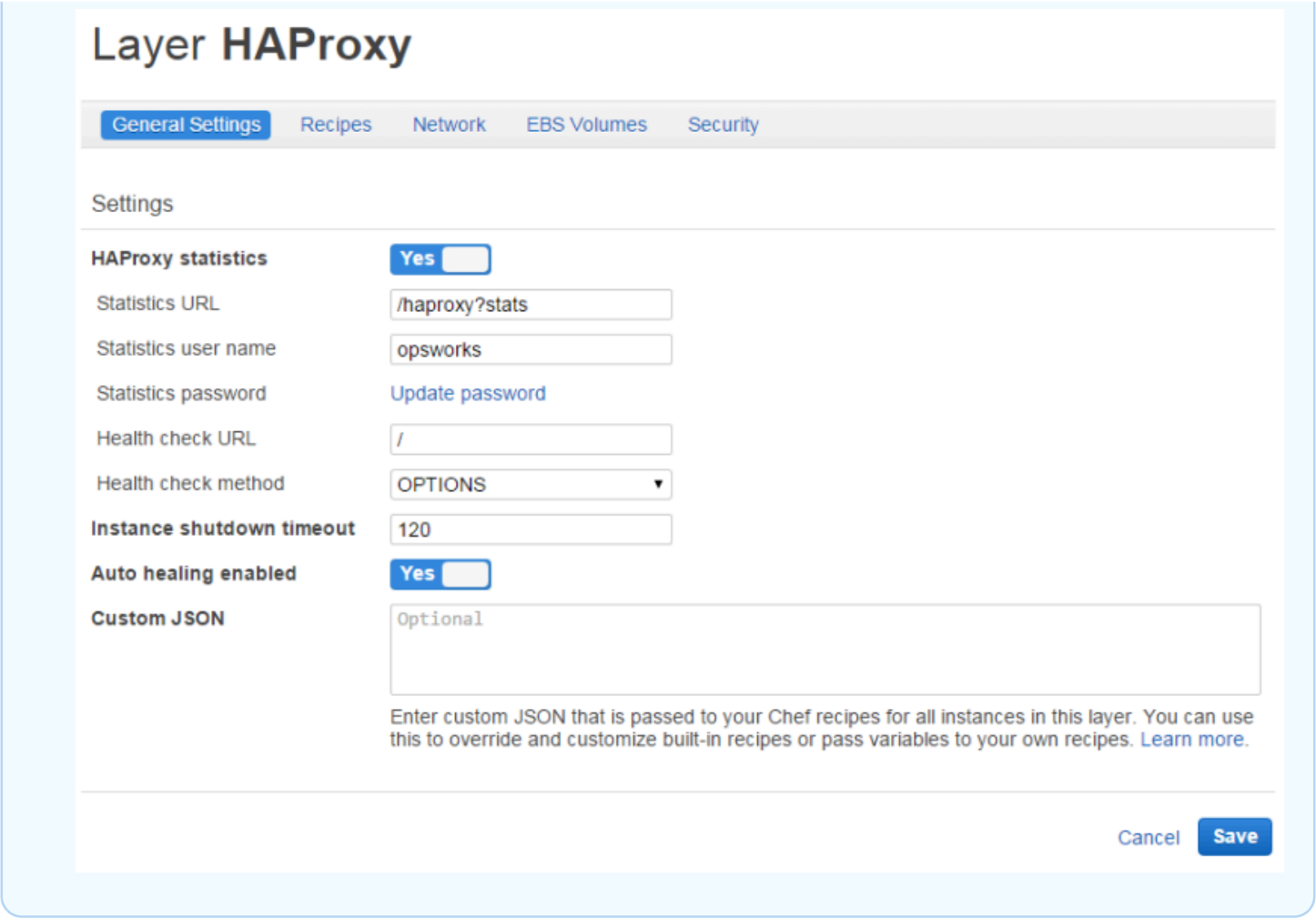
Gruppi di sicurezza personalizzati

Questa impostazione viene visualizzata se si sceglie di non associare automaticamente un gruppo di sicurezza AWS OpsWorks Stacks integrato ai livelli. Devi specificare il gruppo di sicurezza da associare al livello. Assicurati che il gruppo abbia le impostazioni corrette per consentire il traffico tra livelli. Per ulteriori informazioni, consulta Creare un nuovo stack.



Note

Registra la password per un uso successivo; AWS OpsWorks Stacks non consente di visualizzare la password dopo aver creato il layer. Tuttavia, puoi aggiornare la password. A tale scopo, passa alla pagina Edit (Modifica) del livello e fai clic su Update password (Aggiorna password) nella scheda General Settings (Impostazioni generali).



Come funziona il HAProxy layer

Per impostazione predefinita, HAProxy esegue le seguenti operazioni:

- Rimane in ascolto delle richieste sulle porte HTTP e HTTPS.

  Puoi configurare l' HAProxy ascolto solo sulla porta HTTP o HTTPS sovrascrivendo il modello di configurazione Chef,`haproxy.cfg.erb`.
- Instrada il traffico in entrata alle istanze membro di qualsiasi livello del server di applicazioni.

  Per impostazione predefinita, AWS OpsWorks Stacks si configura HAProxy per distribuire il traffico alle istanze che sono membri di qualsiasi livello di application server. Ad esempio, potresti avere uno stack con entrambi i livelli Rails App Server e PHP App Server e un HAProxy master distribuisce il traffico alle istanze in entrambi i livelli. Puoi configurare l'instradamento predefinito utilizzando una ricetta personalizzata.

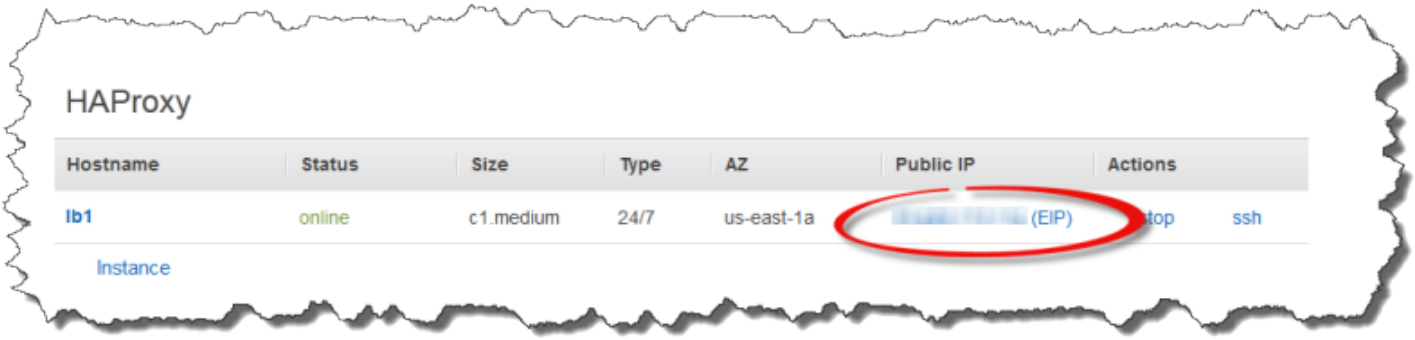
- Instrada il traffico tra più zone di disponibilità.

  Se una zona di disponibilità non è più raggiungibile, il sistema di bilanciamento del carico instrada il traffico in entrata alle istanze in altre zone, in modo che l'applicazione continui a funzionare senza interruzioni. Per questo motivo, si consiglia di distribuire i server di applicazioni su più zone di disponibilità.
- Esegue periodicamente il metodo di controllo dello stato specificato su ogni istanza del server di applicazioni per valutarne lo stato di integrità.

  Se il metodo non viene restituito entro un periodo di timeout specificato, si presume che l'istanza abbia avuto esito negativo e HAProxy interrompe l'instradamento delle richieste verso l'istanza. AWS OpsWorks Stacks fornisce anche un modo per sostituire automaticamente le istanze fallite. Per ulteriori informazioni, consulta Utilizzo della diagnostica automatica. Puoi modificare il metodo di controllo dello stato quando crei il livello.
- Raccoglie le statistiche e, facoltativamente, le visualizza in una pagina Web.

> **Important**
>
> Per garantire il corretto funzionamento del controllo dello stato con il metodo predefinito OPTIONS, l'app deve restituire un codice di stato 2xx o 3xx.

Per impostazione predefinita, quando aggiungi un'istanza a un HAProxy livello, AWS OpsWorks Stacks le assegna un indirizzo IP elastico per rappresentare l'applicazione, che è pubblica a livello mondiale. Poiché l'indirizzo IP elastico (EIP) dell'istanza HAProxy è l'unico URL esposto pubblicamente dell'applicazione, non dovrai creare e gestire nomi di dominio pubblici per le istanze sottostanti dei server di applicazioni. Per recuperare l'indirizzo, passa alla pagina Instances (Istanze) ed esamina l'indirizzo IP pubblico dell'istanza, come illustrato nella seguente figura. Un indirizzo seguito da (EIP) è un indirizzo IP elastico. Per ulteriori informazioni sugli indirizzi IP elastici, consulta l'argomento relativo agli indirizzi IP elastici (EIP).



Quando si arresta un' HAProxy istanza, AWS OpsWorks Stacks conserva l'indirizzo IP elastico e lo riassegna all'istanza al riavvio. Se elimini un' HAProxy istanza, per impostazione predefinita, AWS OpsWorks Stacks elimina l'indirizzo IP dell'istanza. Per conservare l'indirizzo, deseleziona l'opzione Delete instance's Elastic IP (Elimina IP elastico istanza), come illustrato di seguito.



Questa opzione ha effetti su cosa succede quando aggiungi una nuova istanza al livello per sostituire un'istanza eliminata:

- Se hai mantenuto l'indirizzo IP elastico dell'istanza eliminata, AWS OpsWorks Stacks assegna l'indirizzo alla nuova istanza.
- Altrimenti, AWS OpsWorks Stacks assegna un nuovo indirizzo IP elastico all'istanza e devi aggiornare le impostazioni del registrar DNS per mappare il nuovo indirizzo.

Quando le istanze dell'application server sono on line o disattivate, manualmente o come conseguenza del ridimensionamento automatico o della riparazione automatica, la configurazione del bilanciamento del carico deve essere aggiornata per indirizzare il traffico verso il set corrente di istanze online. Questa operazione viene gestita automaticamente dalle ricette predefinite del livello:

- Quando nuove istanze sono online, Stacks attiva un evento Configure lifecycle. AWS OpsWorks Le ricette di configurazione integrate nel HAProxy layer aggiornano la configurazione del load balancer in modo che distribuisca anche le richieste a tutte le nuove istanze dell'Application Server.
- Quando le istanze vanno fuori linea o un'istanza non supera un controllo di integrità, AWS OpsWorks Stacks attiva anche un evento Configure lifecycle. Le ricette HAProxy Configure aggiornano la configurazione del load balancer per indirizzare il traffico solo verso le restanti istanze online.

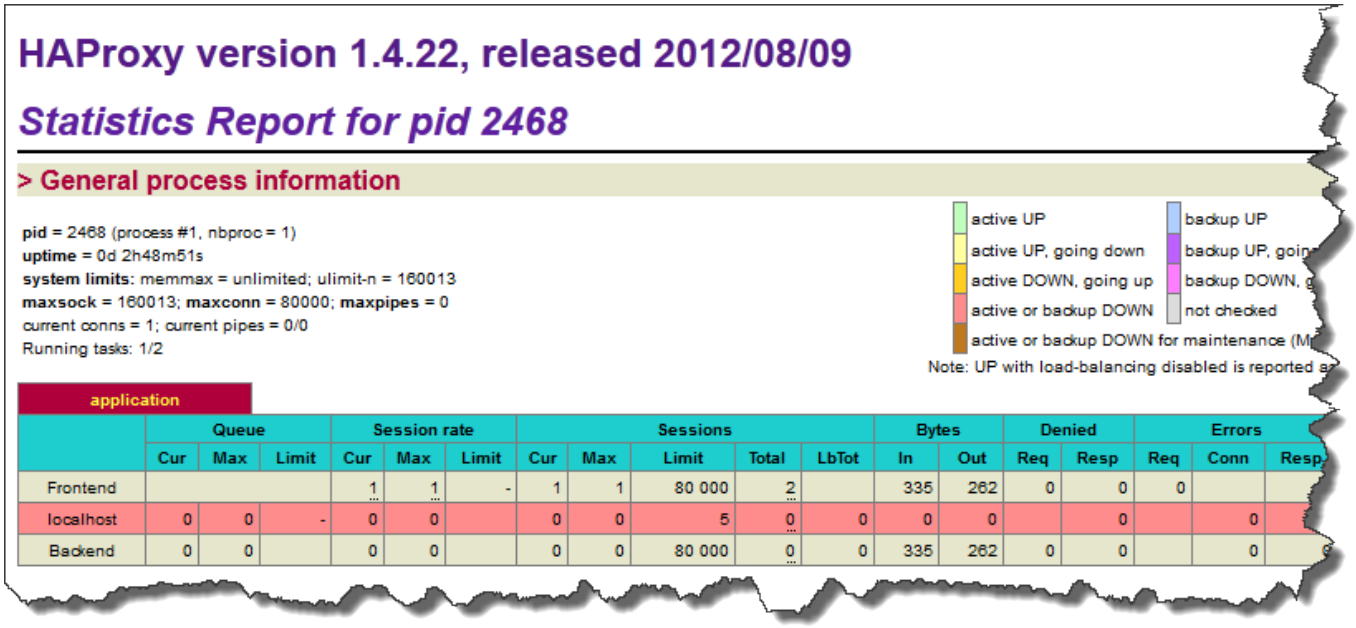
Infine, puoi anche utilizzare un dominio personalizzato con il HAProxy layer. Per ulteriori informazioni, consulta Utilizzo di domini personalizzati.

Pagina delle statistiche

Se hai abilitato la pagina delle statistiche, HAProxy viene visualizzata una pagina contenente una varietà di metriche all'URL specificato.

Per visualizzare le statistiche HAProxy

1. Ottieni il nome DNS pubblico dell' HAProxy istanza dalla pagina dei dettagli dell'istanza e copialo.
2. Nella pagina Layers (Livelli), fare clic su HAProxy per visualizzare la pagina dei dettagli del livello.
3. Ottieni l'URL delle statistiche dai dettagli del layer e aggiungilo al nome DNS pubblico. Ad esempio, aggiungere **`http://ec2-54-245-102-172.us-west-2.compute.amazonaws.com/haproxy?stats`**.
4. Incollare l'URL recuperato alla fase precedente nel browser e utilizzare il nome utente e la password specificati durante la creazione del livello per visualizzare la pagina delle statistiche.



Riferimento ai livelli MySQL

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Il livello MySQL supporta MySQL, un sistema di gestione di database relazionali ampiamente utilizzato. AWS OpsWorks Stacks installa la versione più recente disponibile, che dipende dal sistema operativo. Se aggiungi un'istanza MySQL, le informazioni di accesso necessarie vengono fornite ai livelli del server di applicazioni. È necessario scrivere ricette Chef personalizzate per configurare configurazioni master-master o master-slave.

Short name (Nome breve): db-master

Compatibilità: un livello MySQL è compatibile con i seguenti livelli: custom, lb, memcached, monitoring-master, nodejs-app, php-app, rails-app e web.

Porte aperte: un livello MySQL consente l'accesso pubblico alla porta 22 (SSH) e a tutte le porte dei server Web dello stack, dei server personalizzati e dei server delle applicazioni Rails, PHP e Node.js.

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): sì, su `/vol/mysql`

AWS-OpsWorks-DB-MasterGruppo di sicurezza predefinito: -Server

Configurazione: per configurare un livello MySQL, è necessario specificare quanto segue:

- (Password utente root
- Motore MySQL

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- mysql::server
- dependencies
- deploy::mysql

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client

- ssh_users
- agent_version
- deploy::mysql

Deploy recipes (Ricette di distribuzione):

- deploy::default
- deploy::mysql

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default
- mysql::stop

Installation (Installazione):

- AWS OpsWorks Stacks utilizza il programma di installazione del pacchetto dell'istanza per installare MySQL e i relativi file di registro nelle posizioni predefinite. Per ulteriori informazioni, consulta la documentazione di MySQL.

Livello MySQL OpsWorks

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Note

Questo livello è disponibile solo per stack Chef 11 e stack basati su Linux di versioni precedenti.

Un livello OpsWorks MySQL fornisce un modello per le istanze EC2 Amazon che funzionano come master di database MySQL. Una ricetta integrata crea un database per ciascuna applicazione distribuita su un livello del server applicazioni. Ad esempio, se distribuisci un'applicazione PHP "myapp", la ricetta crea un database "myapp".

Il livello MySQL ha le seguenti impostazioni di configurazione.

MySQL root user password (Password utente root MySQL)

(Obbligatoria) La password utente root.

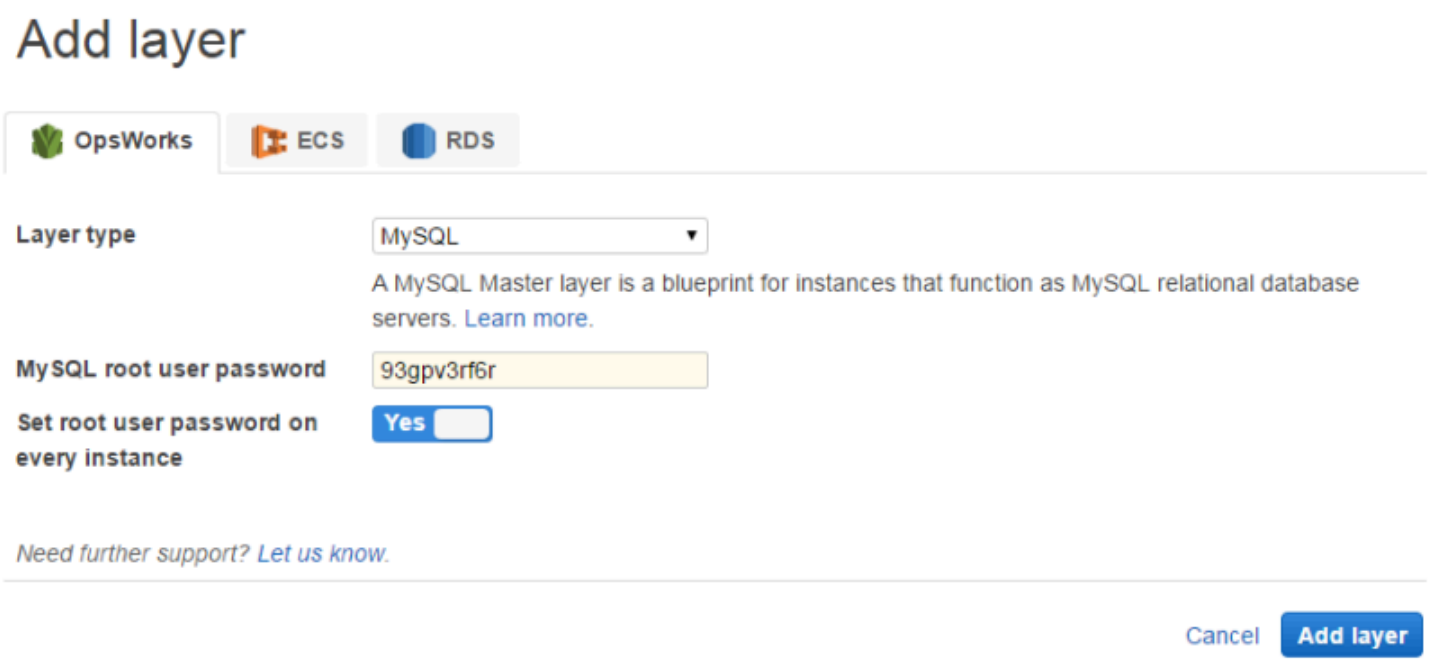
Set root user password on every instance (Imposta la password utente root su ogni istanza)

(Facoltativa) Indica se la password utente root è inclusa negli attributi di configurazione e distribuzione dello stack installati in ogni istanza nello stack. L'impostazione predefinita è Yes (Sì).

Se impostate questo valore su No, AWS OpsWorks Stacks passa la password root solo alle istanze dell'Application Server.

Gruppi di sicurezza personalizzati

(Facoltativa) Gruppo di sicurezza personalizzato da associare al livello. Per ulteriori informazioni, consulta Creare un nuovo stack.



Puoi aggiungere una o più istanze al livello, ciascuna delle quali rappresenta un master del database MySQL separato. Puoi quindi collegare un'istanza a un'app, installando così le informazioni di connessione necessarie sui server applicazioni dell'app. L'applicazione può dunque utilizzare queste informazioni di connessione per connettersi al server di database dell'istanza.

Documentazione di riferimento sul livello del server di applicazioni

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks supporta diversi server di applicazioni e pagine Web statiche.

Argomenti

- Riferimento ai livelli di AWS Flow (Ruby)
- Riferimento al livello di Java App Server
- Riferimento al livello dell'App Server Node.js
- Riferimento al livello di PHP App Server
- Riferimento al livello del server dell'app Rails
- Riferimento statico al livello del server Web

Riferimento ai livelli di AWS Flow (Ruby)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Un layer AWS Flow (Ruby) fornisce un modello per le istanze che ospitano attività e workflow worker di Amazon Simple Workflow Service.

Nome breve: aws-flow-ruby

Compatibilità: un livello AWS Flow (Ruby) è compatibile con PHP App Server, MySQL, Memcached, Ganglia e livelli personalizzati.

Open ports (Porte aperte): nessuna.

Ruolo IAM: aws-opsworks-ec 2- role-with-swf è il ruolo AWS Flow (Ruby) standard che AWS OpsWorks Stacks crea per te, se richiesto.

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): no

Gruppo di sicurezza predefinito: -Ruby-Server AWS-OpsWorks-AWS-Flow

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- opsworks_aws_flow_ruby::setup

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client

- ssh_users
- mysql::client
- agent_version
- opsworks_aws_flow_ruby::configure

Deploy recipes (Ricette di distribuzione):

- deploy::default
- distribuire: aws-flow-ruby

Undeploy recipes (Ricette di annullamento distribuzione):

- distribuire: aws-flow-ruby-undeploy

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default

Riferimento al livello di Java App Server

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Un livello Java App Server supporta un server di applicazioni Apache Tomcat 7.0.

Short name (Nome breve): java-app

Compatibilità: un livello Java App Server è compatibile con i seguenti livelli: custom, db-master e memcached.

Porte aperte: un livello Java App Server consente l'accesso pubblico alle porte 22 (SSH), 80 (HTTP), 443 (HTTPS) e a tutte le porte dei sistemi di bilanciamento del carico.

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): no

Gruppo di sicurezza predefinito: -Server AWS-OpsWorks-Java-App

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- opsworks_java::setup

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- opsworks_java::configure

Deploy recipes (Ricette di distribuzione):

- deploy::default
- deploy::java

Undeploy recipes (Ricette di annullamento distribuzione):

- deploy::java-undeploy

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default
- deploy::java-stop

Installation (Installazione):

- Tomcat esegue l'installazione su `/usr/share/tomcat7`.
- Per ulteriori informazioni su come produrre i file di log, consulta l'argomento sul logging in Tomcat.

Riferimento al livello dell'App Server Node.js

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Un livello di Node.js App Server supporta un server di applicazioni Node.js, che è una piattaforma per l'implementazione di server di applicazioni di rete altamente scalabili. I programmi vengono scritti in JavaScript modalità asincrona basata sugli eventi per ridurre al minimo il sovraccarico e I/O massimizzare la scalabilità.

Short name (Nome breve): nodejs-app

Compatibilità: un livello di Node.js App Server è compatibile con i seguenti livelli: custom, db-master, memcached e monitoring-master.

Porte aperte: un livello Node.js App Server consente l'accesso pubblico alle porte 22 (SSH), 80 (HTTP), 443 (HTTPS) e a tutte le porte dei sistemi di bilanciamento del carico.

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): no

Gruppo di sicurezza predefinito: -Server AWS-OpsWorks-nodejs-App

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- opsworks_nodejs
- opsworks_nodejs::npm

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- opsworks_nodejs::configure

Deploy recipes (Ricette di distribuzione):

- deploy::default
- opsworks_nodejs
- opsworks_nodejs::npm

- deploy::nodejs

Undeploy recipes (Ricette di annullamento distribuzione):

- deploy::nodejs-undeploy

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default
- deploy::nodejs-stop

Installation (Installazione):

- Node.js esegue l'installazione su `/usr/local/bin/node`.
- Per ulteriori informazioni su come produrre i file di log, consulta l'argomento sul logging in node.js nel sito web Nodejitsu.

Node.js application configuration (Configurazione dell'applicazione Node.js):

- Il file principale eseguito da Node.js deve essere denominato `server.js` e trovarsi nella directory radice dell'applicazione distribuita.
- L'applicazione Node.js deve essere impostata per l'ascolto sulla porta 80 (o sulla porta 443, se applicabile).

> **Note**
>
> Le app Node.js che eseguono Express di solito utilizzano il codice seguente per impostare la porta di ascolto, dove `process.env.PORT` rappresenta la porta predefinita e risolve su 80:
>
> ```
> app.set('port', process.env.PORT || 3000);
> ```
>
> Con AWS OpsWorks Stacks, è necessario specificare esplicitamente la porta 80, come segue:
>
> ```
> app.set('port', 80);
> ```

Riferimento al livello di PHP App Server

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Il livello PHP App Server supporta un server di applicazioni PHP utilizzando Apache2 con mod_php.

Short name (Nome breve): php-app

Compatibilità: un livello PHP App Server è compatibile con i seguenti livelli: custom, db-master, memcached, monitoring-master e rails-app.

Porte aperte: un livello PHP App Server consente l'accesso pubblico alle porte 22 (SSH), 80 (HTTP), 443 (HTTPS) e a tutte le porte dei sistemi di bilanciamento del carico.

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): no

Gruppo di sicurezza predefinito: -Server AWS-OpsWorks-PHP-App

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies

- ebs
- opsworks_ganglia::client
- mysql::client
- dependencies
- mod_php5_apache2

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- mod_php5_apache2::php
- php::configure

Deploy recipes (Ricette di distribuzione):

- deploy::default
- deploy::php

Undeploy recipes (Ricette di annullamento distribuzione):

- deploy::php-undeploy

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default
- apache2::stop

Installation (Installazione):

- AWS OpsWorks Stacks utilizza il programma di installazione del pacchetto dell'istanza per installare Apache2, mod_php e i file di registro associati nelle posizioni predefinite. Per ulteriori informazioni sull'installazione, consulta Apache. Per ulteriori informazioni sulla registrazione, consulta la pagina relativa ai file di log.

Riferimento al livello del server dell'app Rails

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Il livello Rails App Server supporta un server di applicazioni Ruby on Rails.

Short name (Nome breve): rails-app

Compatibilità: un livello Rails App Server è compatibile con i seguenti livelli: custom, db-master, memcached, monitoring-master, php-app.

Porte: un livello Rails App Server consente l'accesso pubblico alle porte 22 (SSH), 80 (HTTP), 443 (HTTPS) e a tutte le porte dei sistemi di bilanciamento del carico.

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): no

Gruppo di sicurezza predefinito: -Server AWS-OpsWorks-Rails-App

Configurazione: per configurare un livello Rails App Server, è necessario specificare quanto segue:

- Versione di Ruby
- Stack Rails
- Versione Rubygems
- Se installare e gestire Bundler
- La versione Bundler

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- apache2 apache2::mod_deflate
- passenger_apache2
- passenger_apache2::mod_rails
- passenger_apache2::rails

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- rails::configure

Deploy recipes (Ricette di distribuzione):

- deploy::default
- deploy::rails

Undeploy recipes (Ricette di annullamento distribuzione):

- deploy::rails-undeploy

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default
- apache2::stop

Installation (Installazione):

- AWS OpsWorks Stacks utilizza il programma di installazione del pacchetto dell'istanza per installare Apache2 con mod_passenger, mod_rails e i file di registro associati nelle posizioni predefinite. Per ulteriori informazioni sull'installazione, consulta Phusion Passenger. Per ulteriori informazioni sulla registrazione, consulta la pagina relativa ai file di log.

Riferimento statico al livello del server Web

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Il livello Static Web Server fornisce pagine HTML statiche, che possono includere codice lato client come. JavaScript È basato su Nginx, che è un server proxy HTTP open source, di posta elettronica nonché proxy inverso.

Short name (Nome breve): web

Compatibilità: un livello di server Web statico è compatibile con i seguenti livelli: custom, db-master, memcached.

Porte aperte: un livello di server Web statico consente l'accesso pubblico alle porte 22 (SSH), 80 (HTTP), 443 (HTTPS) e a tutte le porte dei sistemi di bilanciamento del carico.

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): no

Gruppo di sicurezza predefinito: AWS-OpsWorks-Web-Server

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- nginx

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version

Deploy recipes (Ricette di distribuzione):

- deploy::default
- deploy::web

Undeploy recipes (Ricette di annullamento distribuzione):

- deploy::web-undeploy

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default
- nginx::stop

Installation (Installazione):

- Nginx esegue l'installazione su `/usr/sbin/nginx`.
- I file di log di Nginx si trovano su `/var/log/nginx`.

Livelli del server di applicazioni

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi livelli sono disponibili solo per stack Chef 11 e stack basati su Linux di versioni precedenti.

AWS OpsWorks Stacks supporta diversi server di applicazioni, dove «applicazione» include pagine Web statiche. Ogni tipo di server ha un livello AWS OpsWorks Stacks separato, con ricette integrate che gestiscono l'installazione del server delle applicazioni e tutti i pacchetti correlati su ciascuna istanza del livello, la distribuzione delle app e così via. Ad esempio, il livello Java App Server installa diversi pacchetti, tra cui Apache, Tomcat e OpenJDK, e distribuisce app Java su ciascuna istanza del livello.

Di seguito è riportata la procedura di base per l'utilizzo dei livelli di un server di applicazioni:

1. Creare uno dei tipi di livello App Server (Server app) disponibili.
2. Aggiungere una o più istanze al livello.
3. Creare app e distribuirle nelle istanze. Per ulteriori informazioni, consulta App.
4. (Facoltativo) Se il livello ha più istanze, è possibile aggiungere un sistema di bilanciamento del carico, che distribuisce il traffico in ingresso tra le istanze. Per ulteriori informazioni, consulta HAProxy AWS OpsWorks Stacks Layer.

Argomenti

- Strato AWS Flow (Ruby)
- Java App Server AWS OpsWorks Stacks Layer
- Node.js App Server AWS OpsWorks Stacks Layer
- PHP App Server AWS OpsWorks Stacks Layer
- Rails AWS OpsWorks App Server Stacks Layer
- Static Web Server AWS OpsWorks Stacks Layer

Strato AWS Flow (Ruby)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Un livello AWS Flow (Ruby) è un livello AWS OpsWorks Stacks che fornisce un modello per le istanze che ospitano i lavoratori di attività e flussi di lavoro di Amazon SWF. I worker vengono implementati utilizzando AWS Flow Framework for Ruby, un framework di programmazione che semplifica il processo di implementazione di un'applicazione asincrona distribuita fornendo al contempo tutti i vantaggi di Amazon SWF. Questo scenario è ideale per implementare applicazioni e soddisfare un'ampia gamma di scenari, tra cui processi aziendali, codifica multimediale, attività con esecuzione prolungata ed elaborazione in background.

Il livello AWS Flow (Ruby) include le seguenti impostazioni di configurazione.

RubyGems versione

La versione Gem del framework.

Bundler version (Versione Bundler)

La versione Bundler.

EC2 profilo dell'istanza

Un profilo di EC2 istanza Amazon definito dall'utente che deve essere utilizzato dalle istanze del layer. Questo profilo deve concedere le autorizzazioni alle applicazioni in esecuzione sulle istanze del layer per accedere ad Amazon SWF.

Se il tuo account non dispone di un profilo appropriato, puoi selezionare Nuovo profilo con accesso SWF per cui AWS OpsWorks Stacks aggiorni il profilo oppure puoi aggiornarlo tu stesso utilizzando la console IAM. Quindi, puoi utilizzare il profilo aggiornato per tutti i livelli AWS Flow successivi. Di seguito è riportata una breve descrizione di come creare il profilo utilizzando la console IAM. Per ulteriori informazioni, consulta Identity and Access Management in Amazon Simple Workflow Service.

Creazione di un profilo per le istanze AWS Flow (Ruby)

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Policy nel riquadro di navigazione e scegli Crea policy per creare una nuova policy gestita dal cliente.
3. Per Service, scegli SWF.
4. Per Azioni, scegliete Tutte le azioni SWF (swf: *).
5. Per Amazon Resource Name (ARN), inserisci l'ARN che specifica a quali domini Amazon SWF possono accedere i lavoratori. Scegli di fornire **`All resources`** l'accesso a tutti i domini.
6. Scegli Next (Successivo).
7. Facoltativamente, inserisci un tag per identificare la politica.
8. Scegli Next (Successivo).
9. Quando hai finito, scegli Crea politica.
10. Scegli Ruoli nel riquadro di navigazione e scegli Crea ruolo.
11. Specificate il nome del ruolo e scegliete Passaggio successivo. Non è possibile modificare il nome una volta che il ruolo è stato creato.
12. Scegli il servizio AWS e poi scegli EC2.
13. Scegli Next (Successivo).
14. Dall'elenco delle politiche di autorizzazione, scegli la politica che hai creato in precedenza.

15. Scegli Next (Successivo).
16. Fornire un nome ruolo e selezionare Crea ruolo. Non è possibile modificare il nome una volta che il ruolo è stato creato.
17. Specificate questo profilo quando create un layer AWS Flow (Ruby) in AWS OpsWorks Stacks.

Java App Server AWS OpsWorks Stacks Layer

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Il livello Java App Server è un livello AWS OpsWorks Stacks che fornisce un modello per le istanze che funzionano come server di applicazioni Java. Questo livello è basato su Apache Tomcat 7.0 e Open JDK 7. AWS OpsWorks Stacks installa anche la libreria Java Connector, che consente alle app Java di utilizzare un `DataSource` oggetto JDBC per connettersi a un data store di back-end.

Installazione: Tomcat è installato in `/usr/share/tomcat7`.

La pagina Add Layer (Aggiungi livello) include le opzioni di configurazione seguenti:

Opzioni VM Java

Puoi utilizzare questa impostazione per specificare le opzioni di macchina virtuale Java personalizzate. Non sono disponibili opzioni predefinite. Ad esempio, una serie di opzioni comune è `-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`. Se utilizzi Java VM Options, assicurati di passare un set di opzioni valido; AWS OpsWorks Stacks non convalida la stringa. Se tenti di passare un'opzione non valida, il server Tomcat non si avvia, causando un errore di configurazione. In tal caso, puoi esaminare il log di Chef di configurazione dell'istanza

per conoscere i dettagli. Per ulteriori informazioni su come visualizzare e interpretare i log di Chef, consulta Log di Chef.

Gruppi di sicurezza personalizzati

Questa impostazione viene visualizzata se hai scelto di non associare automaticamente un gruppo di sicurezza AWS OpsWorks Stacks integrato ai tuoi livelli. Devi specificare il gruppo di sicurezza da associare al livello. Per ulteriori informazioni, consulta Creare un nuovo stack.

Elastic Load Balancer

Puoi collegare un load balancer Elastic Load Balancing alle istanze del layer. Per ulteriori informazioni, consulta Elastic Load Balancing Layer.

Puoi specificare altre impostazioni di configurazione utilizzando un JSON personalizzato o un file di attributi personalizzato. Per ulteriori informazioni, consulta Configurazione personalizzata.

> **Important**
>
> Se l'applicazione Java utilizza SSL, si consiglia di disabilitarla, SSLv3 se possibile, per risolvere le vulnerabilità descritte in CVE-2014-3566. Per ulteriori informazioni, consulta Disabilitazione per SSLv3 i server Apache.

Argomenti

- Disabilitazione per SSLv3 i server Apache
- Configurazione personalizzata
- Distribuzione di app Java

Disabilitazione per SSLv3 i server Apache

Per disabilitarlo SSLv3, è necessario modificare l'impostazione del file del server Apache. `ssl.conf` `SSLProtocol` Per fare ciò, è necessario sovrascrivere il file `ssl.conf.erb` modello del libro di cucina apache2 integrato, utilizzato per creare le ricette di installazione del livello Java App Server. `ssl.conf` I dettagli dipendono dal sistema operativo che specifichi per le istanze del livello. Di seguito sono riepilogate le modifiche richieste per i sistemi Amazon Linux e Ubuntu. SSLv3 è disattivato automaticamente per i sistemi Red Hat Enterprise Linux (RHEL). Per ulteriori informazioni su come sostituire un modello integrato, consulta Utilizzo di modelli personalizzati .

Amazon Linux

Il file `ssl.conf.erb` per questi sistemi operativi si trova nella directory `apache2` del libro di ricette `apache2/templates/default/mods`. Di seguito viene mostrata la parte pertinente del file integrato.

```
...
#SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL

# enable only secure protocols: SSLv3 and TLSv1.2, but not SSLv2
SSLProtocol all -SSLv2
</IfModule>
```

Sostituisci `ssl.conf.erb` e modifica l'impostazione `SSLProtocol` come segue.

```
...
#SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL

# enable only secure protocols: SSLv3 and TLSv1.2, but not SSLv2
SSLProtocol all -SSLv3 -SSLv2
</IfModule>
```

Ubuntu 14.04 LTS

Il file `ssl.conf.erb` per questo sistema operativo si trova nella directory `apache2` del libro di ricette `apache2/templates/ubuntu-14.04/mods`. Di seguito viene mostrata la parte pertinente del file integrato.

```
...
# The protocols to enable.
# Available values: all, SSLv3, TLSv1.2
# SSL v2 is no longer supported
SSLProtocol all
...
```

Modifica questa impostazione come segue.

```
...
# The protocols to enable.
# Available values: all, SSLv3, TLSv1.2
# SSL v2 is no longer supported
SSLProtocol all -SSLv3 -SSLv2
...
```

Configurazione personalizzata

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks espone impostazioni di configurazione aggiuntive come attributi incorporati, che si trovano tutti nel namespace. `opsworks_java` Puoi utilizzare un JSON personalizzato o un file di attributi personalizzato per sostituire gli attributi integrati e specificare i valori personalizzati. Ad esempio, le versioni di JVM e Tomcat sono rappresentate dagli attributi `jvm_version` e `java_app_server_version` integrati, entrambi impostati su 7. Puoi utilizzare un JSON personalizzato o un file di attributi personalizzato che dovrai impostare (uno o entrambi) su 6. L'esempio seguente utilizza un JSON personalizzato per impostare entrambi gli attributi sul valore 6:

```
{
  "opsworks_java": {
    "jvm_version": 6,
    "java_app_server_version" : 6
  }
}
```

Per ulteriori informazioni, consulta Utilizzo di un JSON personalizzato.

Un altro esempio di configurazione personalizzata consiste nell'installare un JDK personalizzato sostituendo gli attributi `use_custom_pkg_location`, `custom_pkg_location_url_debian` e `custom_pkg_location_url_rhel`.

> **Note**
>
> Se sostituisci i libri di ricette integrati, dovrai aggiornare personalmente questi componenti.

Per ulteriori informazioni sugli attributi e su come sostituirli, consulta Sostituzione degli attributi. Per un elenco degli attributi integrati, consulta Attributi opsworks_java.

Distribuzione di app Java

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

I seguenti argomenti descrivono come distribuire app su istanze di un livello di Java App Server. Gli esempi riguardano le app JSP, ma puoi utilizzare essenzialmente la stessa procedura per installare anche altri tipi di app Java.

Puoi distribuire pagine JSP da uno dei repository supportati. Se desideri distribuire file WAR, tieni presente che AWS OpsWorks Stacks estrae automaticamente i file WAR distribuiti da un archivio Amazon S3 o HTTP, ma non da un repository Git o Subversion. Se desideri usare Git o Subversion per i file WAR, puoi procedere in uno dei seguenti modi:

- Archivia l'archivio estratto nel repository.
- Archivia il file WAR nel repository e utilizza un hook di distribuzione Chef per estrarre l'archivio, come descritto nell'esempio seguente.

Puoi utilizzare gli hook di distribuzione Chef per eseguire applicazioni Ruby fornite dall'utente su un'istanza in una qualsiasi delle quattro fasi di implementazione. Il nome dell'applicazione determina la fase. Di seguito è riportato un esempio di applicazione Ruby denominato `before_migrate.rb`, che estrae un file WAR distribuito da un repository Git o Subversion. Il nome associa l'applicazione all'hook di distribuzione Checkout affinché venga eseguito all'inizio dell'operazione di distribuzione,

dopo la verifica del codice, ma prima della migrazione. Per ulteriori informazioni su come usare questo esempio, consulta Utilizzo degli hook di distribuzione di Chef.

```
::Dir.glob(::File.join(release_path, '*.war')) do |archive_file|
  execute "unzip_#{archive_file}" do
    command "unzip #{archive_file}"
    cwd release_path
  end
end
```

> **Note**
>
> Quando distribuisci un aggiornamento per un'app JSP, Tomcat potrebbe non riconoscere l'aggiornamento e continuare invece a eseguire la versione esistente dell'app. Questo può verificarsi, ad esempio, se distribuisci l'app come file ZIP contenente solo una pagina JSP. Per assicurarti che Tomcat esegua la versione distribuita più recente, la directory root del progetto deve includere una directory WEB-INF contenente un file `web.xml`. Un file `web.xml` può includere vari tipi di contenuto, ma quello riportato di seguito è sufficiente a garantire che Tomcat riconosca gli aggiornamenti ed esegua la versione dell'app attualmente distribuita. Non devi modificare la versione per ogni aggiornamento. Tomcat riconoscerà l'aggiornamento anche se la versione non è cambiata.
>
> ```
> <context-param>
>   <param-name>appVersion</param-name>
>   <param-value>0.1</param-value>
> </context-param>
> ```

Argomenti

- Distribuzione di un'app JSP
- Distribuzione di un'app JSP con un database di back-end

Distribuzione di un'app JSP

Per distribuire un'app JSP, specifica il nome e le informazioni relative al repository. Puoi specificare anche i domini e le impostazioni SSL. Per ulteriori informazioni su come creare un'app, consulta

Aggiunta di app. La procedura seguente mostra come creare e distribuire una semplice pagina JSP da un archivio pubblico di Amazon S3. Per informazioni su come utilizzare altri tipi di repository, inclusi gli archivi privati di Amazon S3, consulta. Origine dell'applicazione

L'esempio seguente mostra la pagina JSP in cui sono visualizzate semplicemente alcune informazioni di sistema.

```
<%@ page import="java.net.InetAddress" %>
<html>
<body>
<%
    java.util.Date date = new java.util.Date();
    InetAddress inetAddress = InetAddress.getLocalHost();
%>
The time is
<%
    out.println( date );
    out.println("<br>Your server's hostname is "+inetAddress.getHostName());
%>
<br>
</body>
</html>
```

> **Note**
>
> La procedura seguente presuppone una buona familiarità con le nozioni di base di creazione degli stack, di aggiunta ai livelli e così via. Se non conosci AWS OpsWorks Stacks, dovresti prima vedere. Nozioni di base sugli stack Linux Chef 11

Per distribuire una pagina JSP da un archivio Amazon S3

1. Crea uno stack con un livello Java App Server, aggiungi un'istanza 24 ore su 24, 7 giorni su 7 al livello e avvialo.
2. Copiare il codice in un file denominato `simplejsp.jsp`, inserire il file in una cartella chiamata `simplejsp` e creare un archivio `.zip` della cartella. I nomi sono arbitrari. È possibile utilizzare qualsiasi nome file o cartella. È inoltre possibile utilizzare altri tipi di archivio, tra cui i file gzip, bzip2, tarball o Java WAR. Nota che AWS OpsWorks Stacks non supporta archivi tar non compressi. Per distribuire più pagine JSP, includerle nello stesso archivio.

3. Carica l'archivio in un bucket Amazon S3 e rendi pubblico il file. Copiare l'URL del file per utilizzarlo in futuro. Per ulteriori informazioni su come creare bucket e caricare file, consulta Nozioni di base su Amazon Simple Storage Service.
4. Aggiungere un'app allo stack e specificare le impostazioni seguenti:

   - Nome: `SimpleJSP`
   - App type (Tipo di app) - `Java`
   - Repository type (Tipo di repository) - `Http Archive`
   - URL del repository: l'URL Amazon S3 del tuo file di archivio.

   Utilizzare i valori predefiniti delle impostazioni restanti e fare clic su Add App (Aggiungi app) per creare l'app.
5. Distribuisci l'app nell'istanza di Java App Server.

Adesso, puoi accedere all'URL dell'app e visualizzarla. Se non hai specificato un dominio, puoi creare un URL utilizzando l'indirizzo IP pubblico dell'istanza o il relativo nome DNS pubblico. Per ottenere l'indirizzo IP pubblico o il nome DNS pubblico di un'istanza, vai alla console AWS OpsWorks Stacks e fai clic sul nome dell'istanza nella pagina Istanze per aprirne la pagina dei dettagli.

Il resto dell'URL dipende dal nome breve dell'app, che è un nome minuscolo generato da AWS OpsWorks Stacks dal nome dell'app che hai specificato al momento della creazione dell'app. Ad esempio, il nome breve di SimpleJSP è simplejsp. Puoi ottenere il nome breve di un'app dalla relativa pagina dei dettagli.

- Se il nome breve è `root`, puoi utilizzare `http://`*`public_DNS`*`/`*`appname`*`.jsp` o `http://`*`public_IP`*`/`*`appname`*`.jsp`.
- In alternativa, puoi utilizzare `http://`*`public_DNS`*`/`*`app_shortname`*`/`*`appname`*`.jsp` o `http://`*`public_IP`*`/`*`app_shortname`*`/`*`appname`*`.jsp`.

Se hai specificato un dominio per l'app, l'URL è `http://`*`domain`*`/`*`appname`*`.jsp`.

L'URL dell'esempio dovrebbe essere simile a `http://192.0.2.0/simplejsp/simplejsp.jsp`.

Se desideri distribuire più app nella stessa istanza, non devi usare `root` come nome breve. Questo può causare conflitti dell'URL in grado di impedire il corretto funzionamento dell'app. Al contrario, assegna un altro nome di dominio a ogni app.

Distribuzione di un'app JSP con un database di back-end

Le pagine JSP possono utilizzare un oggetto `DataSource` JDBC per la connessione a un database di back-end. Puoi creare e distribuire tale app utilizzando la procedura nella sezione precedente, con un passaggio aggiuntivo per configurare la connessione.

La seguente pagina JSP mostra come connettersi a un oggetto `DataSource`.

```
<html>
  <head>
    <title>DB Access</title>
  </head>
  <body>
    <%@ page language="java" import="java.sql.*,javax.naming.*,javax.sql.*" %>
    <%
      StringBuffer output = new StringBuffer();
      DataSource ds = null;
      Connection con = null;
      Statement stmt = null;
      ResultSet rs = null;
      try {
        Context initCtx = new InitialContext();
        ds = (DataSource) initCtx.lookup("java:comp/env/jdbc/mydb");
        con = ds.getConnection();
        output.append("Databases found:<br>");
        stmt = con.createStatement();
        rs = stmt.executeQuery("show databases");
        while (rs.next()) {
          output.append(rs.getString(1));
          output.append("<br>");
        }
      }
      catch (Exception e) {
        output.append("Exception: ");
        output.append(e.getMessage());
        output.append("<br>");
      }
      finally {
        try {
          if (rs != null) {
            rs.close();
          }
```

```
          if (stmt != null) {
            stmt.close();
          }
          if (con != null) {
            con.close();
          }
        }
        catch (Exception e) {
          output.append("Exception (during close of connection): ");
          output.append(e.getMessage());
          output.append("<br>");
        }
      }
    %>
    <%= output.toString() %>
  </body>
</html>
```

AWS OpsWorks Stacks crea e inizializza l'`DataSource`oggetto, lo associa a un nome logico e lo registra con un servizio di denominazione JNDI (Java Naming and Directory Interface). Il nome logico completo è `java:comp/env/`*`user-assigned-name`*. Devi specificare la parte del nome assegnata dall'utente aggiungendo attributi JSON personalizzati agli attributi di configurazione e distribuzione dello stack per definire l'attributo `['opsworks_java']['datasources']`, come descritto di seguito.

Per distribuire una pagina JSP che si connette a un database MySQL

1. Crea uno stack con un livello Java App Server, aggiungi istanze 24 ore su 24, 7 giorni su 7 a ciascun livello e avvialo.
2. Aggiungere un livello di database allo stack. I dettagli dipendono dal database utilizzato.

   Per utilizzare un'istanza MySQL per l'esempio, aggiungi un livello MySQL allo stack, aggiungi un'istanza 24 ore su 24, 7 giorni su 7 al livello e avvialo.

   Per utilizzare un'istanza Amazon RDS (MySQL) per l'esempio:

   - Specificare un motore di database MySQL per l'istanza.
   - Assegna i gruppi di sicurezza AWS- OpsWorks -DB-master-Server () ***`security_group_id`*** e AWS- OpsWorks -Java-App-Server () all'istanza. *`security_group_id`* AWS OpsWorks Stacks crea questi gruppi di sicurezza per te quando crei il tuo primo stack nella regione.

- Creare un database denominato `simplejspdb`.
- Verificare che il nome utente e la password master non contengano & o altri caratteri che potrebbero causare un errore di Tomcat.

  In particolare durante l'avvio, Tomcat deve analizzare il file di contesto dell'app Web, ovvero un file XML che include il nome utente e la password master. Se una delle stringhe include un carattere &, il parser XML la tratta come un'entità XML danneggiata e genera un'eccezione di analisi che impedisce l'avvio di Tomcat. Per ulteriori informazioni sul file di contesto dell'app Web, consulta tomcat::context.
- Aggiungi un driver MySQL al livello Java App Server.
- Registrare l'istanza RDS con lo stack.

Per ulteriori informazioni su come utilizzare le istanze Amazon RDS con AWS OpsWorks Stacks, consulta. Amazon RDS Service Layer

3. Copiare il codice di esempio in un file denominato `simplejspdb.jsp`, inserire il file in una cartella chiamata `simplejspdb` e creare un archivio `.zip` della cartella. I nomi sono arbitrari. È possibile utilizzare qualsiasi nome file o cartella. È inoltre possibile utilizzare altri tipi di archivio, tra cui i file gzip, bzip2 o tarball. Per distribuire più pagine JSP, includerle nello stesso archivio. Per informazioni su come distribuire app da altri tipi di repository, consulta Origine dell'applicazione.
4. Carica l'archivio in un bucket Amazon S3 e rendi pubblico il file. Copiare l'URL del file per utilizzarlo in futuro. Per ulteriori informazioni su come creare bucket e caricare file, consulta Nozioni di base su Amazon Simple Storage Service.
5. Aggiungere un'app allo stack e specificare le impostazioni seguenti:

   - Nome: `SimpleJSPDB`
   - App type (Tipo di app) - `Java`
   - Tipo di origine dati: OpsWorks(per un'istanza MySQL) o RDS (per un'istanza Amazon RDS).
   - Istanza di database: l'istanza MySQL creata in precedenza, che in genere è denominata db-master1 (mysql), o l'istanza Amazon RDS, che verrà denominata (mysql). ***`DB_instance_name`***
   - Database name (Nome del database) - `simplejspdb`.
   - Repository type (Tipo di repository) - `Http Archive`
   - URL del repository: l'URL Amazon S3 del tuo file di archivio.

Utilizzare i valori predefiniti delle impostazioni restanti e fare clic su Add App (Aggiungi app) per creare l'app.

6. Aggiungere i seguenti attributi JSON personalizzati agli attributi di configurazione dello stack, dove simplejspdb è il nome breve dell'app.

```
{
  "opsworks_java": {
    "datasources": {
      "simplejspdb": "jdbc/mydb"
    }
  }
}
```

AWS OpsWorks Stacks utilizza questa mappatura per generare un file di contesto con le informazioni di database necessarie.

Per ulteriori informazioni su come aggiungere attributi JSON personalizzati agli attributi di configurazione dello stack, consulta Utilizzo di un JSON personalizzato.

7. Distribuisci l'app nell'istanza di Java App Server.

Adesso, puoi utilizzare l'URL dell'app per visualizzarla. Per informazioni su come creare l'URL, consulta Distribuzione di un'app JSP.

L'URL dell'esempio dovrebbe essere simile a `http://192.0.2.0/simplejspdb/simplejspdb.jsp`.

> **Note**
>
> L'attributo `datasources` può contenere più attributi. Ogni attributo viene denominato con il nome breve di un'app e impostato sulla parte appropriata, assegnata dall'utente, di un nome logico. Se disponi di più app, puoi utilizzare nomi logici separati che richiedono un JSON personalizzato analogo a quello riportato di seguito.
>
> ```
> {
>   "opsworks_java": {
>     "datasources": {
> ```

```
      "myjavaapp": "jdbc/myappdb",
      "simplejsp": "jdbc/myjspdb",
      ...
    }
  }
}
```

Node.js App Server AWS OpsWorks Stacks Layer

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Il livello Node.js App Server è un livello AWS OpsWorks Stacks che fornisce un modello per le istanze che funzionano come server di applicazioni Node.js. AWS OpsWorks Stacks installa anche Express, quindi le istanze del layer supportano sia le applicazioni standard che quelle Express.

Installazione: Node.js è installato in `/usr/local/bin/node`.

La pagina Add Layer (Aggiungi livello) include le opzioni di configurazione seguenti:

Node.js version (Versione Node.js)

Per un elenco delle versioni attualmente supportate, consulta AWS OpsWorks Sistemi operativi impilati.

Gruppi di sicurezza personalizzati

Questa impostazione viene visualizzata se hai scelto di non associare automaticamente un gruppo di sicurezza AWS OpsWorks Stacks integrato ai tuoi livelli. Devi specificare il gruppo di sicurezza da associare al livello. Per ulteriori informazioni, consulta Creare un nuovo stack.

Elastic Load Balancer

Puoi collegare un load balancer Elastic Load Balancing alle istanze del layer.

> **Important**
>
> Se l'applicazione Node.js utilizza SSL, si consiglia di disabilitarla, SSLv3 se possibile, per risolvere le vulnerabilità descritte in CVE-2015-8027. Per farlo, devi impostare Node.js version (Versione Node.js) su `0.12.9`.

Distribuzione di app Node.js

Per informazioni dettagliate sull'implementazione di un'applicazione Node.js semplice per AWS OpsWorks Stacks e sulla relativa distribuzione in uno stack, consulta Creazione del primo stack Node.js. In generale, le applicazioni Node.js per AWS OpsWorks Stacks devono soddisfare le condizioni seguenti:

- Il file principale deve essere denominato `server.js` e trovarsi nella directory radice dell'applicazione distribuita.
- Le app Express devono includere un file `package.json` nella directory radice dell'applicazione.
- Per impostazione predefinita, l'applicazione deve essere in ascolto sulla porta 80 (HTTP) o 443 (HTTPS).

  È possibile ascoltare su altre porte, ma il gruppo di sicurezza integrato del livello di Node.js App Server, AWS- OpsWorks -NodeJS-App-Server, consente il traffico utente in entrata solo verso le porte 80, 443 e 22 (SSH). Per consentire il traffico utente in entrata verso altre porte, crea un gruppo di sicurezza con regole in entrata appropriate e assegnalo al livello Node.js App Server. Non modificare le regole in entrata modificando il gruppo di sicurezza predefinito. Ogni volta che crei uno stack, AWS OpsWorks Stacks sovrascrive i gruppi di sicurezza integrati con le impostazioni standard, quindi tutte le modifiche apportate andranno perse.

> **Note**
>
> AWS OpsWorks Stacks imposta la variabile di ambiente PORT su 80 (impostazione predefinita) o 443 (se abiliti SSL), in modo da poter utilizzare il codice seguente per ascoltare le richieste.
>
> ```
> app.listen(process.env.PORT);
> ```

Se configuri un'app Node.js per supportare SSL, devi specificare la chiave e i certificati. AWS OpsWorks Stacks inserisce i dati per ogni istanza del server delle applicazioni come file separati nella `/srv/www/app_shortname/shared/config` directory, come segue.

- `ssl.crt`— il certificato SSL.
- `ssl.key`— la chiave SSL.
- `ssl.ca`— il certificato della catena, se ne hai specificato uno.

La tua applicazione è in grado di ottenere la chiave SSL e i certificati da tali file.

PHP App Server AWS OpsWorks Stacks Layer

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Il livello PHP App Server è un livello AWS OpsWorks Stacks che fornisce un modello per le istanze che funzionano come server di applicazioni PHP. Il livello PHP App Server è basato su Apache2 con e non ha opzioni di configurazione standard. mod_php La versione PHP e Apache dipende dal sistema operativo che specifichi per le istanze del livello.

Sistema operativo	Versione PHP	Versione Apache
Amazon Linux 2018.03	5.3	2.2
Amazon Linux 2017.09	5.3	2.2
Amazon Linux 2017.03	5.3	2.2
Amazon Linux 2016.09	5.3	2.2
Amazon Linux 2016.03	5.3	2.2
Amazon Linux 2015.09	5.3	2.2
Amazon Linux 2015.03	5.3	2.2
Amazon Linux 2014.09	5.3	2.2
Ubuntu 14.04 LTS	5.5	2.4

Installazione: AWS OpsWorks Stacks utilizza il programma di installazione del pacchetto dell'istanza per installare Apache2 e nelle relative posizioni predefinite. mod_php Per ulteriori informazioni sull'installazione, consulta Apache.

La pagina Add Layer (Aggiungi livello) include le opzioni di configurazione seguenti:

Gruppi di sicurezza personalizzati

Questa impostazione viene visualizzata se scegli di non associare automaticamente un gruppo di sicurezza AWS OpsWorks Stacks integrato ai tuoi livelli. Devi specificare il gruppo di sicurezza da associare al livello. Per ulteriori informazioni, consulta Creare un nuovo stack.

Elastic Load Balancer

Puoi collegare un load balancer Elastic Load Balancing alle istanze del layer.

Puoi modificare alcune impostazioni di configurazione di Apache utilizzando un JSON personalizzato o un file di attributi personalizzato. Per ulteriori informazioni, consulta Sostituzione degli attributi. Per un elenco degli attributi Apache che è possibile sostituire, consulta Attributi apache2.

Per ottenere un esempio di come distribuire un'app PHP, nonché di come connettere l'app a un database di back-end, consulta Nozioni di base sugli stack Linux Chef 11.

> **Important**
>
> Se la tua applicazione PHP utilizza SSL, ti consigliamo di disabilitarla, SSLv3 se possibile, per risolvere le vulnerabilità descritte in CVE-2014-3566. A tale scopo, devi modificare l'impostazione `SSLProtocol` nel file `ssl.conf` del server Apache. Per ulteriori informazioni su come modificare questa impostazione, consulta Disabilitazione per SSLv3 i server Apache.

Rails AWS OpsWorks App Server Stacks Layer

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Il livello Rails App Server è un livello AWS OpsWorks Stacks che fornisce un modello per le istanze che funzionano come server di applicazioni Rails.

Installazione: AWS OpsWorks Stacks utilizza il programma di installazione dei pacchetti dell'istanza per installare i pacchetti del server nelle loro posizioni predefinite. Per ulteriori informazioni sull' Apache/Passenger installazione, consulta Phusion Passenger. Per ulteriori informazioni sulla

registrazione, consulta la pagina relativa ai file di log. Per ulteriori informazioni sull' Nginx/Unicorn installazione, consulta Unicorn.

La pagina Add Layer (Aggiungi livello) include le opzioni di configurazione facoltative seguenti.

Ruby Version (Versione Ruby)

La versione di Ruby che verrà utilizzata dalle applicazioni. Il valore predefinito è 2.3.

Puoi anche specificare la tua versione preferita di Ruby sostituendo l'attributo `[:opsworks][:ruby_version]`.

> **Note**
>
> AWS OpsWorks Stacks installa un pacchetto Ruby separato che può essere utilizzato dalle ricette e dall'agent di istanza. Per ulteriori informazioni, consulta Versioni di Ruby.

Stack Rails

Lo stack Rails predefinito è Apache2 con Phusion Passenger. Puoi utilizzare anche Nginx con Unicorn.

> **Note**
>
> Se utilizzi Nginx e Unicorn, devi aggiungere il gem unicorn al Gemfile dell'app, come illustrato nell'esempio seguente:
>
> ```
> source 'https://rubygems.org'
> gem 'rails', '3.2.15'
> ...
> # Use unicorn as the app server
> gem 'unicorn'
> ...
> ```

Passenger Version (Versione Passenger)

Se hai specificato Apache2/Passenger, devi specificare la versione di Passenger. Il valore predefinito è 5.0.28.

Rubygems Version (Versione Rubygems)

La versione predefinita di Rubygems è la 2.5.1.

Install and Manage Bundler (Installa e gestisci Bundler)

Consente di scegliere se installare e gestire Bundler. Il valore predefinito è Yes (Sì).

Bundler version (Versione Bundler)

La versione predefinita di Bundler è la 1.12.5.

Gruppi di sicurezza personalizzati

Questa impostazione viene visualizzata se hai scelto di non associare automaticamente un gruppo di sicurezza AWS OpsWorks Stacks integrato ai tuoi livelli. Devi specificare il gruppo di sicurezza da associare al livello. Per ulteriori informazioni, consulta Creare un nuovo stack.

Elastic Load Balancer

Puoi collegare un load balancer Elastic Load Balancing alle istanze del layer.

Puoi modificare alcune impostazioni di configurazione utilizzando un JSON personalizzato o un file di attributi personalizzato. Per ulteriori informazioni, consulta Sostituzione degli attributi. Per un elenco degli attributi Apache, Nginx, Passenger e Unicorn Phusion che possono essere sostituiti, consulta Attributi dei libri di ricette predefiniti.

> **Important**
>
> Se l'applicazione Ruby on Rails utilizza SSL, si consiglia di disabilitarla, SSLv3 se possibile, per risolvere le vulnerabilità descritte in CVE-2014-3566. Per ulteriori informazioni, consulta Disabilitazione per i server Rails SSLv3 .

Argomenti

- Disabilitazione per i server Rails SSLv3
- Connessione a un database
- Distribuzione di app Ruby on Rails

Disabilitazione per i server Rails SSLv3

Per disabilitarla SSLv3 per i server Rails, aggiorna l'impostazione della versione Ruby del layer alla 2.1 o superiore, che installa Ruby 2.1.4 o superiore come versione utilizzata dalle applicazioni.

- Aggiorna l'impostazione Ruby Version (Versione Ruby) del livello su 2.1 o versione successiva.
- Aggiorna il file di configurazione per il tuo stack Rails, come segue.

Apache con Phusion Passenger

Aggiorna l'impostazione `SSLProtocol` nel file `ssl.conf` del server Apache, come descritto in Disabilitazione per SSLv3 i server Apache.

Nginx con Unicorn

Aggiungi una direttiva `ssl_protocols` esplicita al file `nginx.conf` del server Nginx. Per disabilitarlo SSLv3, sovrascrivi il file modello `nginx.conf.erb` del ricettario nginx integrato, che le ricette di configurazione del livello Rails App Server utilizzano per `nginx.conf` creare, e aggiungi la seguente direttiva:

```
ssl_protocols TLSv1.2;
```

Per ulteriori informazioni su come configurare `nginx.conf`, consulta la pagina relativa alla configurazione di server HTTPS. Per ulteriori informazioni su come sostituire un modello integrato, consulta Utilizzo di modelli personalizzati .

Connessione a un database

Quando distribuisci un'app, AWS OpsWorks Stacks crea un nuovo `database.yml` file utilizzando le informazioni degli attributi dell'app. `deploy` Se colleghi un'istanza MySQL o Amazon RDS all' AWS OpsWorks app, Stacks aggiunge le informazioni `deploy` di connessione agli attributi, in `database.yml` modo che contengano automaticamente i dati di connessione corretti.

Se un'app non ha un database allegato, per impostazione predefinita, AWS OpsWorks Stacks non aggiunge alcuna informazione di connessione agli `deploy` attributi e non crea. `database.yml` Se desideri utilizzare un database diverso, puoi usare un JSON personalizzato per aggiungere attributi di database agli attributi `deploy` dell'app con le informazioni di connessione. Gli attributi sono tutti sotto`["deploy"]["`*`appshortname`*`"]["database"]`, *`appshortname`* dov'è il nome

breve dell'app, che AWS OpsWorks Stacks genera dal nome dell'app. I valori specificati in un JSON personalizzato sostituiscono le impostazioni predefinite. Per ulteriori informazioni, consulta Aggiunta di app.

AWS OpsWorks Stacks incorpora i seguenti valori di `[:...][:database]`attributo in. `database.yml` Gli attributi richiesti dipendono dal particolare database, ma è necessario disporre di un `host` attributo altrimenti AWS OpsWorks Stacks non verrà creato. `database.yml`

- `[:adapter] (String)`— L'adattatore del database, ad esempio`mysql`.
- `[:database]`(String) — Il nome del database.
- `[:encoding]`(String) — La codifica, che in genere è impostata su. `utf8`
- `[:host]`(String) — L'URL dell'host, ad esempio`railsexample.cdlqlk5uwd0k.us-west-2.rds.amazonaws.com`.
- `[:reconnect]`(Boolean) — Indica se l'applicazione deve riconnettersi se la connessione non esiste più.
- `[:password]`(String) — La password del database.
- `[:port]` (numero). — Il numero di porta del database. Utilizza questo attributo per sostituire il numero di porta predefinito, che viene impostato dall'adattatore.
- `[:username]`(String) — Il nome utente del database.

L'esempio seguente mostra un JSON personalizzato per un'app il cui nome breve è myapp.

```
{
  "deploy" : {
    "myapp" : {
      "database" : {
        "adapter" : "adapter",
        "database" : "databasename",
        "host" : "host",
        "password" : "password",
        "port" : portnumber
        "reconnect" : true/false,
        "username" : "username"
      }
    }
  }
}
```

Per ulteriori informazioni su come specificare un JSON personalizzato, consulta Utilizzo di un JSON personalizzato. Per visualizzare il modello utilizzato per creare `database.yml` (`database.yml.erb`), vai al repository del libro di ricette integrato.

Distribuzione di app Ruby on Rails

Puoi distribuire app Ruby on Rails da uno qualsiasi dei repository supportati. Quanto segue mostra come distribuire un'app Ruby on Rails di esempio su un server che esegue uno Apache/Passenger stack Rails. Il codice di esempio è archiviato in un GitHub archivio pubblico, ma la procedura di base è la stessa per gli altri repository supportati. Per ulteriori informazioni su come creare e distribuire app, consulta App. Per visualizzare il codice di esempio, che include commenti dettagliati, vai su - share-app. https://github.com/awslabs/ opsworks-demo-rails-photo

Per distribuire un'app Ruby on Rails da un repository GitHub

1. Crea uno stack con un layer Rails App Server Apache/Passenger come stack Rails, aggiungi un'istanza 24/7 al layer e avvialo.
2. Una volta che l'istanza è online, aggiungere un'app allo stack e specificare le impostazioni seguenti:

   - Name (Nome) - Il nome preferito. Nell'esempio viene utilizzato `PhotoPoll`.

     AWS OpsWorks Stacks utilizza questo nome per scopi di visualizzazione e genera un nome breve per uso interno e per identificare l'app negli attributi di configurazione e distribuzione dello stack. Ad esempio, il nome PhotoPoll breve è photopoll.
   - App type (Tipo di app) - Ruby on Rails.
   - Rails environment (Ambiente Rails) - Gli ambienti disponibili sono determinati dall'applicazione.

     L'app di esempio ne ha tre: **`development`**, **`test`** e **`production`**. In questo esempio impostare l'ambiente su **`development`**. Vedere il codice di esempio per le descrizioni di ciascun ambiente.
   - Tipo di repository: qualsiasi tipo di repository supportato. Specificare `Git` per questo esempio
   - Repository URL (URL repository) - Il repository da cui dovrebbe essere distribuito il codice.

     In questo esempio impostare l'URL su **`git://github.com/awslabs/opsworks-demo-rails-photo-share-app`**.

Utilizzare i valori predefiniti delle impostazioni restanti e fare clic su Add App (Aggiungi app) per creare l'app.

3. Distribuisci l'app sull'istanza di Rails App Server.
4. Al termine della distribuzione, vai alla pagina Istanze e fai clic sull'indirizzo IP pubblico dell'istanza di Rails App Server. Verrà visualizzato un codice analogo al seguente:



Static Web Server AWS OpsWorks Stacks Layer

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Il livello Static Web Server è un livello AWS OpsWorks Stacks che fornisce un modello per le istanze per la pubblicazione di pagine HTML statiche, che possono includere script lato client. Questo livello è basato su Nginx.

Installazione: Nginx è installato in `/usr/sbin/nginx`.

La pagina Add Layer (Aggiungi livello) include le opzioni di configurazione seguenti:

Gruppi di sicurezza personalizzati

Questa impostazione viene visualizzata se si sceglie di non associare automaticamente un gruppo di sicurezza AWS OpsWorks Stacks integrato ai livelli. Devi specificare il gruppo di sicurezza da associare al livello. Per ulteriori informazioni, consulta Creare un nuovo stack.

Elastic Load Balancer

Puoi collegare un load balancer Elastic Load Balancing alle istanze del layer.

Puoi modificare alcune impostazioni di configurazione di Nginx utilizzando un JSON personalizzato o un file di attributi personalizzato. Per ulteriori informazioni, consulta Sostituzione degli attributi. Per un elenco degli attributi Apache che è possibile sostituire, consulta Attributi nginx.

> **Important**
>
> Se la tua applicazione web utilizza SSL, ti consigliamo di disabilitarla, SSLv3 se possibile, per risolvere le vulnerabilità descritte in CVE-2014-3566.
> Per disabilitarlo SSLv3, è necessario modificare il file del server Nginx. `nginx.conf` Per fare ciò, sovrascrivi il file modello del ricettario nginx integrato, che le ricette `nginx.conf.erb` di configurazione del livello Rails App Server utilizzano per `nginx.conf` creare, e aggiungi la seguente direttiva:
>
> ```
> ssl_protocols TLSv1.2;
> ```

Per ulteriori informazioni su come configurare `nginx.conf`, consulta la pagina relativa alla configurazione di server HTTPS. Per ulteriori informazioni su come sostituire un modello integrato, consulta Utilizzo di modelli personalizzati .

Riferimento al livello del cluster ECS

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Un livello ECS Cluster rappresenta un cluster Amazon Elastic Container Service (Amazon ECS) e semplifica la gestione del cluster.

Short name (Nome breve): ecs-cluster

Compatibilità: un livello di servizio Amazon ECS è compatibile solo con livelli personalizzati

Porte aperte: ECS Cluster consente l'accesso pubblico alla porta 22 (SSH)

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): no

Gruppo di sicurezza predefinito: AWS-OpsWorks-ECS-Cluster

Configurazione: per configurare un livello di cluster ECS, è necessario specificare quanto segue:

- Se assegnare automaticamente indirizzi IP pubblici o indirizzi IP elastici alle istanze di container
- Il profilo dell'istanza per l'istanza di container

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users

- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- opsworks_ecs::setup

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client
- ssh_users
- mysql::client
- agent_version
- opsworks_ecs::configure

Deploy recipes (Ricette di distribuzione):

- deploy::default
- opsworks_ecs::deploy

Undeploy recipes (Ricette di annullamento distribuzione):

- opsworks_ecs::undeploy

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default
- opsworks_ecs::shutdown

Installation (Installazione):

- AWS OpsWorks Stacks utilizza il programma di installazione del pacchetto dell'istanza per installare Docker nelle posizioni predefinite
- Il registro Chef per l'evento Setup rileva se l'agente Amazon ECS è stato installato correttamente. Altrimenti, i log forniti da AWS OpsWorks Stacks non includono informazioni sui log degli errori

di Amazon ECS. Per ulteriori informazioni su come gestire gli errori di Amazon ECS, consulta Amazon ECS Troubleshooting.

Documentazione di riferimento sui livelli personalizzati

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se i livelli standard non rispondono alle specifiche esigenze, puoi creare un livello personalizzato. Uno stack può avere più livelli personalizzati. Per impostazione predefinita, il livello personalizzato esegue un gruppo limitato di ricette standard che supportano le funzionalità di base. Le funzionalità principali del livello vengono quindi implementate tramite un set di ricette Chef personalizzate per ciascun evento del ciclo di vita appropriato per impostare e configurare il software del livello e così via. Le ricette personalizzate vengono aggiunte alle ricette AWS OpsWorks Stacks standard per ogni evento.

Short name (Nome breve): definito dall'utente; ciascun livello personalizzato di uno stack deve avere un nome breve diverso

Open ports (Porte aperte): per impostazione predefinita, un livello server personalizzato apre l'accesso pubblico alle porte 22 (SSH), 80 (HTTP), 443 (HTTPS) e a tutte le porte dei livelli dei server di applicazioni Rails e PHP dello stack

Autoassign Elastic IP Addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): no

Gruppo di sicurezza predefinito: AWS-OpsWorks-Custom-Server

Compatibility (Compatibilità): i livelli personalizzati sono compatibili con i seguenti livelli: personalizzato, db-master, lb, memcached, monitoring-master, nodejs-app, php-app, rails-app e web

Configuration (Configurazione): per configurare un livello personalizzato, è necessario specificare quanto segue:

- Il nome del livello
- Il nome breve del livello, che identifica quest'ultimo nelle ricette Chef e devono includere solo lettere (a-z) e numeri

Per gli stack Linux, il livello personalizzato utilizza le seguenti ricette.

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version

Deploy recipes (Ricette di distribuzione):

- deploy::default

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default

Documentazione di riferimento su altri livelli

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks supporta anche i seguenti livelli.

Argomenti

- Ganglia Layer Reference
- Riferimento al livello Memcached

Ganglia Layer Reference

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Un livello Ganglia supporta Ganglia, un sistema di monitoraggio distribuito che gestisce l'archiviazione e la visualizzazione delle metriche delle istanze. È concepito per funzionare con topologie di istanza gerarchiche, il che lo rende particolarmente utile per gruppi di istanze. Ganglia dispone di due componenti di base:

- Un client a sovraccarico ridotto, che è installato in ogni istanza dello stack e invia i parametri al master.
- Un master, che raccoglie le metriche dei clienti e le archivia su un volume Amazon EBS. Visualizza inoltre i parametri su una pagina Web.

AWS OpsWorks Stacks dispone di un agente di monitoraggio Ganglia su ogni istanza gestita. Quando aggiungi un layer Ganglia allo stack e lo avvii, gli agenti Ganglia di ogni istanza segnalano le metriche all'istanza Ganglia. Per usare Ganglia, aggiungi un livello Ganglia con un'istanza allo stack. I dati sono accessibili tramite il back-end di Ganglia presso l'indirizzo IP del master. Puoi fornire ulteriori definizioni per i parametri scrivendo ricette Chef.

Short name (Nome breve): monitoring-master

Compatibilità: un livello Ganglia è compatibile con i seguenti livelli: custom, db-master, memcached, php-app, rails-app.

Open ports (Porte aperte): Load-Balancer consente l'accesso pubblico alle porte 22 (SSH), 80 (HTTP) e 443 (HTTPS).

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): sì, su `/vol/ganglia`

AWS-OpsWorks-Monitoring-MasterGruppo di sicurezza predefinito: -Server

Configurazione: per configurare un layer Ganglia, è necessario specificare quanto segue:

- L'URI che fornisce l'accesso ai grafici di monitoraggio. Il valore predefinito è http://*DNSName*/ganglia, dove *DNSName* è il nome DNS dell'istanza Ganglia.
- Un nome utente e password che controllano l'accesso alle statistiche di monitoraggio.

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies

- ebs
- opsworks_ganglia::client
- opsworks_ganglia::server

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- opsworks_ganglia::configure-server

Deploy recipes (Ricette di distribuzione):

- deploy::default
- opsworks_ganglia::configure-server
- opsworks_ganglia::deploy

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default
- apache2::stop

Installation (Installazione):

- Il client di Ganglia è installato in: `/etc/ganglia`.
- Il front-end Web di Ganglia è installato in: `/usr/share/ganglia-webfrontend`.
- Il logtailer di Ganglia è installato in: `/usr/share/ganglia-logtailer`.

Riferimento al livello Memcached

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack basati su Linux.

Memcached è un sistema distribuito di caching su memoria per dati arbitrari. Velocizza i siti Web eseguendo il caching nella RAM di stringhe e oggetti come chiavi e valori per ridurre il numero di volte in cui un'origine dati esterna deve essere letta.

Per utilizzare Memcached in uno stack, crea un layer Memcached e aggiungi una o più istanze, che funzionano come server Memcached. Le istanze installano automaticamente Memcached e le altre istanze dello stack sono in grado di accedere e utilizzare i server Memcached. Se si utilizza un livello Rails App Server, AWS OpsWorks Stacks inserisce automaticamente un file di configurazione nella directory di `memcached.yml` configurazione di ogni istanza del livello. È possibile ottenere il server Memcached e il numero di porta da questo file.

Short name (Nome breve): memcached

Compatibilità: un layer Memcached è compatibile con i seguenti livelli: custom, db-master, lb, monitoring-master, nodejs-app, php-app, rails-app e web.

Porte aperte: un livello Memcached consente l'accesso pubblico alla porta 22 (SSH) e a tutte le porte dei server Web dello stack, dei server personalizzati e dei server delle applicazioni Rails, PHP e Node.js.

Autoassign Elastic IP addresses (Assegnazione automatica indirizzi IP elastici): disattivata per impostazione predefinita

Default EBS volume (Volume EBS predefinito): no

Gruppo di sicurezza predefinito: AWS-OpsWorks-Memcached-Server

Per configurare un layer Memcached, è necessario specificare la dimensione della cache, in MB.

Setup recipes (Ricette di impostazione):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- memcached

Configure recipes (Ricette di configurazione):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version

Deploy recipes (Ricette di distribuzione):

- deploy::default

Shutdown recipes (Ricette di arresto):

- opsworks_shutdown::default
- memcached::stop

Installation (Installazione):

- AWS OpsWorks Stacks utilizza il programma di installazione del pacchetto dell'istanza per installare Memcached e i relativi file di registro nelle posizioni predefinite.

Altri livelli

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi livelli sono disponibili solo per stack Chef 11 e stack basati su Linux di versioni precedenti.

AWS OpsWorks Stacks supporta anche i layer Ganglia e Memcached.

Argomenti

- Ganglia Layer
- Memcached

Ganglia Layer

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack Chef 11 e stack basati su Linux di versioni precedenti.

AWS OpsWorks Stacks invia tutte le metriche relative all'istanza e al volume ad Amazon CloudWatch, semplificando la visualizzazione di grafici e l'impostazione di allarmi per aiutarti a risolvere i problemi e intraprendere azioni automatiche in base allo stato delle tue risorse. Puoi

anche utilizzare il layer Ganglia AWS OpsWorks Stacks per ulteriori opzioni di monitoraggio delle applicazioni, come la memorizzazione delle metriche scelte.

Il livello Ganglia è un modello per un'istanza che monitora lo stack utilizzando il monitoraggio distribuito di Ganglia. Uno stack di solito ha una sola istanza di Ganglia. Il livello Ganglia include le seguenti impostazioni di configurazione opzionali:

Ganglia URL (URL Ganglia)

Percorso dell'URL della pagina delle statistiche. L'URL completo è http://*DNSName*/*URLPath*, dove *DNSName* è il nome DNS dell'istanza associata. Il *URLPath* valore predefinito è «/ganglia» che corrisponde a qualcosa come: http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/ganglia.

Ganglia user name (Nome utente Ganglia)

Nome utente per la pagina Web relativa alle statistiche. È necessario fornire il nome utente quando si visualizza la pagina. Il valore predefinito è «opsworks».

Ganglia password (Password Ganglia)

Password che controlla l'accesso alle pagina Web relativa alle statistiche. Devi fornire la password quando visualizzi la pagina. Il valore predefinito è una stringa generata casualmente.

> **Note**
>
> Registra la password per un uso successivo; AWS OpsWorks Stacks non consente di visualizzare la password dopo aver creato il layer. Tuttavia, puoi aggiornare la password. A tale scopo, passa alla pagina Edit (Modifica) del livello e fai clic su Update password (Aggiorna password).

Gruppi di sicurezza personalizzati

Questa impostazione viene visualizzata se si sceglie di non associare automaticamente un gruppo di sicurezza AWS OpsWorks Stacks integrato ai livelli. Devi specificare il gruppo di sicurezza da associare al livello. Per ulteriori informazioni, consulta Creare un nuovo stack.

Elastic Load Balancer

Puoi collegare un load balancer Elastic Load Balancing alle istanze del layer.

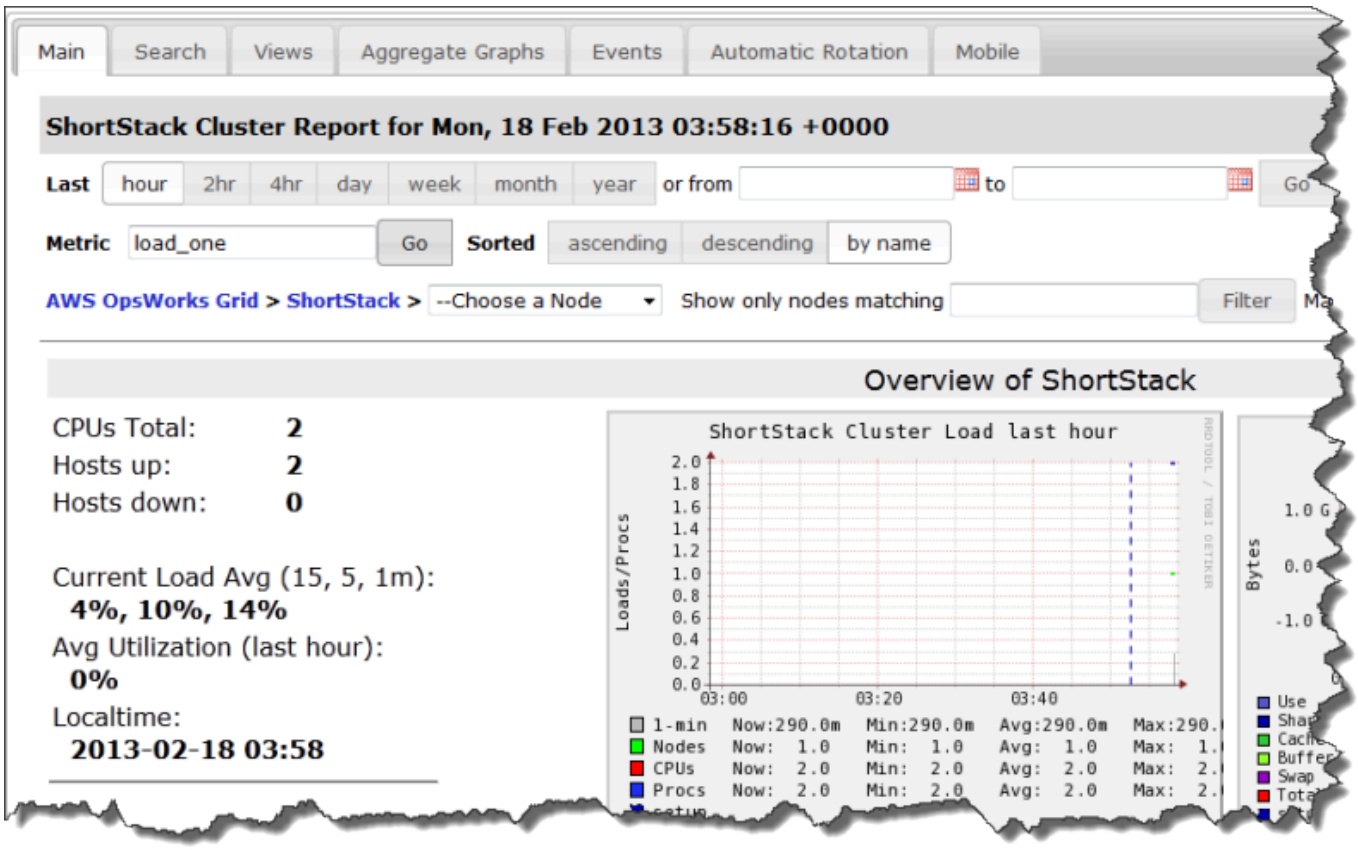
> **Important**
>
> Se lo stack include un layer Ganglia, ti consigliamo di disabilitarlo, SSLv3 se possibile, per risolvere le vulnerabilità descritte in CVE-2014-3566. A tale scopo, devi sostituire il modello `ssl.conf.erb` del server Apache per modificare l'impostazione `SSLProtocol`. Per informazioni dettagliate, consultare Disabilitazione per SSLv3 i server Apache.

Visualizza le statistiche di Ganglia

AWS OpsWorks Le ricette Stacks installano un client Ganglia a basso costo su ogni istanza. Se lo stack include un livello Ganglia, il client Ganglia inizia automaticamente a segnalare a Ganglia non appena l'istanza è online. Ganglia utilizza i dati del client per calcolare una serie di statistiche e visualizza i risultati graficamente sulla sua pagina web dedicata alle statistiche.

Per visualizzare le statistiche Ganglia

1. Nella pagina Layer, fate clic su Ganglia per aprire la pagina dei dettagli del layer.
2. Nel riquadro di navigazione fare clic su Instances (Istanze). In Ganglia, fate clic sul nome dell'istanza.
3. Copiare il nome dell'istanza visualizzato nel campo Public DNS (DNS pubblico).
4. Usa il nome DNS per costruire l'URL delle statistiche, che sarà simile a: http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/ganglia.
5. Incollare l'URL completo nel browser, passare alla pagina corrispondente e immettere il nome utente e la password Ganglia per visualizzare la pagina. Di seguito è riportato un esempio.



Memcached

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo livello è disponibile solo per stack Chef 11 e stack basati su Linux di versioni precedenti.

Un livello Memcached è un livello AWS OpsWorks Stacks che fornisce un modello per le istanze che funzionano come server Memcached, un sistema distribuito di memorizzazione nella cache della memoria per dati arbitrari. Il livello Memcached include le seguenti impostazioni di configurazione.

Allocated memory (MB) (Memoria allocata (MB))

(Opzionale) Quantità di memoria cache, espressa in MB, per ogni istanza del livello. Il valore predefinito è 512 MB.

Gruppi di sicurezza personalizzati

Questa impostazione viene visualizzata se si sceglie di non associare automaticamente un gruppo di sicurezza AWS OpsWorks Stacks integrato ai livelli. Devi specificare il gruppo di sicurezza da associare al livello. Per ulteriori informazioni, consulta Creare un nuovo stack.

Componenti Cookbook

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Un libro di ricette in genere include i componenti di base seguenti:

- I file di attributo contengono una serie di attributi che rappresentano i valori da utilizzare dalle ricette e dai modelli.
- I file di modello sono modelli che le ricette utilizzano per creare altri file, come per esempio i file di configurazione.

  I file modello in genere consentono di modificare il file di configurazione sovrascrivendo gli attributi, operazione che può essere eseguita senza toccare il ricettario, invece di riscrivere un file di configurazione. La pratica standard è che ogni volta che si prevede di modificare un file di configurazione su un'istanza anche solo leggermente, bisogna utilizzare un file di modello.

- I file di ricetta sono applicazioni Ruby che definiscono tutto ciò che è necessario per configurare un sistema, incluse la creazione e la configurazione di cartelle, l'installazione e la configurazione di pacchetti, l'avvio di servizi e così via.

I libri di ricette non devono disporre di tutti e tre i componenti. Gli approcci più semplici alla personalizzazione richiedono solo file di attributo o di modello. Inoltre, i libri di ricette possono includere eventualmente altri tipi di file, come per esempio definizioni o specifiche.

Questa sezione descrive i tre componenti standard del libro di ricette. Per ulteriori informazioni, soprattutto su come implementare le ricette, consulta Opscode.

Argomenti

- Attributes
- Modelli
- Ricette

Attributes

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ricette e modelli dipendono da vari valori, come per esempio le impostazioni di configurazione. Anziché impostare come hardcoded tali valori direttamente nelle ricette o nei modelli, è possibile creare un file di attributo con un attributo che rappresenta ogni valore. È quindi possibile utilizzare gli attributi nelle proprie ricette o nei propri modelli anziché valori espliciti. Il vantaggio di utilizzare gli attributi è che è possibile sostituire i relativi valori senza toccare il libro di ricette. Per questo motivo, bisogna sempre utilizzare gli attributi per definire i seguenti tipi di valori:

- I valori che potrebbero variare da stack a stack o con il tempo, come per esempio i nomi utente.

Se si impostano come hardcoded tali valori, si deve modificare la ricetta o il modello ogni volta che bisogna modificare un valore. Utilizzando gli attributi per definire questi valori, è possibile utilizzare gli stessi libri di ricette per ogni stack e sostituire semplicemente gli attributi appropriati.

- I valori sensibili, come per esempio le password o le chiavi segrete.

  Inserire espliciti valori sensibili nel proprio libro di ricette può aumentare il rischio di esposizione. Al contrario, definire gli attributi con valori fittizi e sostituirli per impostare i valori effettivi. Il modo migliore per sostituire questi attributi è con JSON personalizzati. Per ulteriori informazioni, consulta Utilizzo di un JSON personalizzato.

Per ulteriori informazioni sugli attributi e su come sostituirli, consulta Sostituzione degli attributi.

L'esempio seguente è una parte di un file di attributo di esempio.

```
...
default["apache"]["listen_ports"] = [ '80','443' ]
default["apache"]["contact"] = 'ops@example.com'
default["apache"]["timeout"] = 120
default["apache"]["keepalive"] = 'Off'
default["apache"]["keepaliverequests"] = 100
default["apache"]["keepalivetimeout"] = 3
default["apache"]["prefork"]["startservers"] = 16
default["apache"]["prefork"]["minspareservers"] = 16
default["apache"]["prefork"]["maxspareservers"] = 32
default["apache"]["prefork"]["serverlimit"] = 400
default["apache"]["prefork"]["maxclients"] = 400
default["apache"]["prefork"]["maxrequestsperchild"] = 10000
...
```

AWS OpsWorks Stacks definisce gli attributi utilizzando la seguente sintassi:

```
node.type["attribute"]["subattribute"]["..."]=value
```

È possibile anche utilizzare i due punti (:) come segue:

```
node.type[:attribute][:subattribute][:...]=value
```

Una definizione di attributo presenta i seguenti componenti:

`node.`

Il prefisso `node.` è opzionale e in genere omesso, come indicato nell'esempio.

`type`

Il tipo determina se l'attributo può essere sovrascritto. AWS OpsWorks Gli attributi Stacks utilizzano in genere uno dei seguenti tipi:

- `default` è la tipologia più comunemente utilizzata, in quanto consente all'attributo di essere sostituito.
- `normal`definisce un attributo che sostituisce uno dei valori degli attributi AWS OpsWorks Stacks standard.

> **Note**
>
> Chef supporta tipi aggiuntivi, che non sono necessari per AWS OpsWorks Stacks ma potrebbero essere utili per il tuo progetto. Per ulteriori informazioni, consulta la sezione relativa alle informazioni sugli attributi.

`attribute name`

Il nome attributo utilizza la sintassi standard di nodo Chef, `[:`*`attribute`*`][:`*`subattribute`*`][...]`. È possibile utilizzare qualsiasi nome per gli attributi. Tuttavia, come illustrato in Sostituzione degli attributi, gli attributi personalizzati del libro di ricette vengono uniti nell'oggetto del nodo dell'istanza, assieme agli attributi dalla configurazione dello stack e agli attributi di distribuzione e lo strumento Ohai dello Chef. I nomi di configurazione comunemente utilizzati, come per esempio porta o utente potrebbero comparire in vari libri di ricette.

Per evitare conflitti di nome, la convenzione è di creare nomi di attributo qualificati con almeno due elementi, come indicato nell'esempio. Il primo elemento deve essere unico e in genere è basato su un nome di prodotto, come Apache. È seguito da uno o più sottoattributi che identificano il valore specifico, come per esempio `[:user]` o `[:port]`. È possibile utilizzare tutti i sottoattributi appropriati per il progetto.

`value`

Un attributo può essere impostato per i seguenti tipi di valori:

- Una stringa, come per esempio `default[:apache][:keepalive] = 'Off'`.
- Un numero (senza virgolette), come per esempio `default[:apache][:timeout] = 120`.
- Il valore booleano, che può essere `true` o `false` (senza virgolette).
- Un elenco di valori, come per esempio `default[:apache][:listen_ports] = [ '80','443' ]`.

Il file di attributo è un'applicazione Ruby, quindi è possibile anche utilizzare la sintassi del nodo e gli operatori logici per assegnare valori basati su altri attributi. Per ulteriori informazioni su come definire gli attributi, consulta la sezione relativa alle Informazioni sugli attributi. Per esempi di file di attributi funzionanti, consulta i libri di cucina integrati in AWS OpsWorks Stacks su opsworks-cookbooks. https://github.com/aws/

Modelli

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

È possibile configurare molti pacchetti creando un file di configurazione e posizionandolo nella directory appropriata. È possibile includere un file di configurazione nel proprio libro di ricette e copiarlo nella directory appropriata, ma un approccio più flessibile è far creare alle proprie ricette il file di configurazione da un modello. Uno dei vantaggi di un modello è che è possibile utilizzare gli attributi per definire i valori del modello. In questo modo è possibile, per esempio, modificare un file di configurazione senza toccare il libro di cucina utilizzando JSON personalizzato per sostituire i valori di attributo appropriati.

Un modello ha essenzialmente lo stesso contenuto e la stessa struttura del file associato. Ecco un file di esempio, `httpd.conf`.

```
ServerRoot "<%= node[:apache][:dir] %>"
<% if node[:platform] == "debian" || node[:platform] == "ubuntu" -%>
  LockFile /var/lock/apache2/accept.lock
```

```
<% else -%>
   LockFile logs/accept.lock
<% end -%>
PidFile <%= node[:apache][:pid_file] %>
Timeout <%= node[:apache][:timeout] %>
KeepAlive <%= node[:apache][:keepalive] %>
MaxKeepAliveRequests <%= node[:apache][:keepaliverequests] %>
KeepAliveTimeout <%= node[:apache][:keepalivetimeout] %>
<IfModule mpm_prefork_module>
    StartServers          <%= node[:apache][:prefork][:startservers] %>
    MinSpareServers       <%= node[:apache][:prefork][:minspareservers] %>
    MaxSpareServers       <%= node[:apache][:prefork][:maxspareservers] %>
    ServerLimit           <%= node[:apache][:prefork][:serverlimit] %>
    MaxClients            <%= node[:apache][:prefork][:maxclients] %>
    MaxRequestsPerChild   <%= node[:apache][:prefork][:maxrequestsperchild] %>
</IfModule>
...
```

L'esempio seguente è il file `httpd.conf` che è stato generato per un'istanza Ubuntu:

```
ServerRoot "/etc/httpd"
LockFile logs/accept.lock
PidFile /var/run/httpd/httpd.pid
Timeout 120
KeepAlive Off
MaxKeepAliveRequests 100
KeepAliveTimeout 3
<IfModule mpm_prefork_module>
    StartServers          16
    MinSpareServers       16
    MaxSpareServers       32
    ServerLimit           400
    MaxClients            400
    MaxRequestsPerChild   10000
</IfModule>
...
```

La maggior parte del testo del modello viene semplicemente copiata dal modello al file `httpd.conf`. Tuttavia, il contenuto `<%= ... %>` viene gestito come segue:

- Chef sostituisce `<%= node[:attribute][:sub_attribute][:...]%>` con il valore dell'attributo.

Per esempio, `StartServers <%= node[:apache][:prefork][:startservers] %>` diventa `StartServers 16` nel `httpd.conf`.

- È possibile utilizzare `<%if-%>, <%else-%>, and <%end-%>` per selezionare un valore in base a condizioni.

  L'esempio imposta un percorso di file diverso per `accept.lock` a seconda della piattaforma.

> **Note**
>
> Non si è limitati agli attributi nei file di attributo del proprio libro di ricette. È possibile utilizzare qualsiasi attributo nell'oggetto nodo dell'istanza. Per esempio, generato da uno strumento Chef chiamato Ohai e anche integrato nell'oggetto nodo. Per ulteriori informazioni sugli attributi, consulta Sostituzione degli attributi.

Per ulteriori informazioni sui modelli e su come incorporare il codice Ruby, consulta About Templates (Riguardo agli attributi).

Ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le ricette sono applicazioni Ruby che definiscono una configurazione di sistema. Esse installano pacchetti, creano file di configurazione dai modelli, eseguono comandi shell, creano file e directory e così via. In genere, gli AWS OpsWorks Stacks eseguono le ricette automaticamente quando si verifica un evento del ciclo di vita sull'istanza, ma è anche possibile eseguirle in modo esplicito in qualsiasi momento utilizzando il comando Execute Recipes stack. Per ulteriori informazioni, consulta About Recipes (Informazioni sulle ricette).

Un ricetta in genere consiste in gran parte in una serie di risorse, ciascuna delle quali rappresenta lo stato desiderato di un aspetto del sistema. Ogni risorsa include una serie di attributi che definiscono lo stato desiderato e specificano l'operazione da compiere. Chef associa ogni risorsa a un provider appropriato che compie l'operazione. Per ulteriori informazioni, consulta Resources and Providers Reference (Riferimenti risorse e provider).

Una risorsa `package` aiuta a gestire pacchetti software sulle istanze Linux. L'esempio seguente installa il pacchetto Apache.

```
...
package 'apache2' do
  case node[:platform]
  when 'centos','redhat','fedora','amazon'
    package_name 'httpd'
  when 'debian','ubuntu'
    package_name 'apache2'
  end
  action :install
end
...
```

Chef utilizza il provider di pacchetto appropriato per la piattaforma. Gli attributi di risorsa hanno spesso solo assegnato un valore, ma è possibile utilizzare le operazioni logiche Ruby per eseguire incarichi condizionali. L'esempio utilizza un operatore `case` che utilizza `node[:platform]` per identificare il sistema operativo dell'istanza e imposta l'attributo `package_name` di conseguenza. È possibile inserire gli attributi in una ricetta utilizzando la sintassi standard di nodo Chef e Chef la sostituisce con il valore associato. È possibile utilizzare qualsiasi attributo nell'oggetto nodo, non solo gli attributi del libro di ricette.

Dopo aver determinato il nome di pacchetto appropriato, il segmento di codice termina con un'azione `install` che installa il pacchetto. Altre azioni per questa risorsa includono `upgrade` e `remove`. Per ulteriori informazioni, consulta package.

Spesso è utile dividere attività di installazione e configurazione complesse in una o più sottoattività, ognuna implementata come una ricetta separata e farle eseguire alla ricetta primaria al momento opportuno. L'esempio seguente mostra la riga di codice che segue l'esempio precedente:

```
include_recipe 'apache2::service'
```

Per fare in modo che una ricetta esegua una ricetta figlia, utilizzare la parola chiave `include_recipe` seguita dal nome della ricetta. Le ricette vengono identificate utilizzando la sintassi Chef standard *`CookbookName`*`::`*`RecipeName`*, dove *`RecipeName`* omette l'estensione `.rb`.

> **Note**
>
> Una dichiarazione `include_recipe` esegue in modo efficace la ricetta in quel punto della ricetta primaria. Tuttavia, ciò che effettivamente accade è che Chef sostituisce ogni dichiarazione `include_recipe` con il codice della ricetta specificato prima di eseguire la ricetta primaria.

Una risorsa `directory` rappresenta una directory, come per esempio quella che conterrà i file di un pacchetto. La seguente risorsa `default.rb` crea una directory di log Linux.

```
directory node[:apache][:log_dir] do
    mode 0755
    action :create
end
```

La directory di log è definita in uno dei file di attributo del libro di ricette. La risorsa specifica la modalità di directory come 0755 e utilizza un'operazione `create` per creare la directory. Per ulteriori informazioni, consulta directory. È anche possibile utilizzare questa risorsa con le istanze Windows.

La risorsa `execute` rappresenta i comandi, come per esempio comandi shell o script. L'esempio seguente genera file module.load.

```
execute 'generate-module-list' do
  if node[:kernel][:machine] == 'x86_64'
    libdir = 'lib64'
  else
    libdir = 'lib'
  end
  command "/usr/local/bin/apache2_module_conf_generate.pl /usr/#{libdir}/httpd/
modules /etc/httpd/mods-available"
  action :run
end
```

La risorsa prima determina la tipologia di CPU. `[:kernel][:machine]` è un altro degli attributi automatici che Chef genera per rappresentare varie proprietà di sistema, in questo caso la tipologia di CPU. Poi specifica il comando, uno script Perl e utilizza un'operazione `run` per eseguire lo script, che genera il file module.load. Per ulteriori informazioni, consulta execute.

Una `template` risorsa rappresenta un file, in genere un file di configurazione, che deve essere generato da uno dei file modello del libro di cucina. L'esempio seguente crea un file di configurazione `httpd.conf` dal modello `apache2.conf.erb` che è stato illustrato in Modelli.

```
template 'apache2.conf' do
  case node[:platform]
  when 'centos','redhat','fedora','amazon'
    path "#{node[:apache][:dir]}/conf/httpd.conf"
  when 'debian','ubuntu'
    path "#{node[:apache][:dir]}/apache2.conf"
  end
  source 'apache2.conf.erb'
  owner 'root'
  group 'root'
  mode 0644
  notifies :restart, resources(:service => 'apache2')
end
```

La risorsa determina il nome e la posizione del file generato in base al sistema operativo dell'istanza. Poi specifica `apache2.conf.erb` come modello da utilizzare per generare il file e imposta il proprietario, il gruppo e la modalità del file. Esegue l'operazione `notify` per notificare la risorsa `service` che rappresenta il server Apache per riavviare il server. Per ulteriori informazioni, consulta template.

Attributi di configurazione e distribuzione dello stack: Linux

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo argomento include gli attributi di configurazione e distribuzione dello stack più comunemente utilizzati e la relativa sintassi del nodo associata. È organizzato attorno alla struttura dello spazio dei nomi della configurazione dello stack utilizzata dagli stack Linux. Gli stessi nomi di attributo vengono talvolta utilizzati per scopi differenti e possono essere presenti in spazi dei nomi diversi. Ad esempio, `id` può fare riferimento a un ID stack, un ID livello, un ID app e così via. Pertanto, è necessario il nome completo per utilizzare il valore dell'attributo. Un metodo pratico per visualizzare questi dati è come oggetto JSON. Per alcuni esempi, consultare Attributi di configurazione e distribuzione dello stack.

> **Note**
>
> Sulle istanze Linux, AWS OpsWorks Stacks installa questo oggetto JSON su ogni istanza oltre ad aggiungere i dati all'oggetto nodo. Per recuperarlo devi utilizzare il comando `get_json` dell'interfaccia a riga di comando dell'agente.

Argomenti

- Attributi opsworks
- Attributi opsworks_custom_cookbooks
- Attributi dependencies
- Attributi ganglia
- Attributi mysql
- Attributi passenger
- Attributi opsworks_bundler
- Attributi deploy
- Altri attributi di primo livello

Attributi opsworks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

L'`opsworks`elemento, a volte denominato `opsworks` namespace, contiene un set di attributi che definiscono la configurazione di base dello stack.

> **Important**
>
> La sostituzione dei valori degli attributi nello spazio dei nomi `opsworks` non è consigliata. Tale operazione può comportare l'esito negativo delle ricette integrate.

Argomenti

- applications
- Attributi instance
- Attributi layers
- Attributi rails_stack
- Attributi stack
- Altri attributi opsworks di primo livello

applications

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Contiene un elenco di oggetti incorporati, uno per ogni app esistente nello stack. Ogni oggetto incorporato contiene i seguenti attributi che descrivono la configurazione dell'applicazione.

> **Note**
>
> La sintassi del nodo generale per questi attributi è riportata di seguito. *i* specifica l'indice di elenco a base zero dell'istanza.
>
> ```
> node["opsworks"]["applications"]["i"]["attribute_name"]
> ```

application_type

Il tipo dell'applicazione (stringa). I valori possibili sono i seguenti:

- php: app PHP
- `rails`: un'app Ruby on Rails
- `java`: un'app Java
- `nodejs`: un'app Node.js
- `web`: una pagina HTML statica
- `other`: tutti gli altri tipi di applicazioni

```
node["opsworks"]["applications"]["i"]["application_type"]
```

nome

Il nome di visualizzazione definito dall'utente, ad esempio `"SimplePHP"` (stringa).

```
node["opsworks"]["applications"]["i"]["name"]
```

slug_name

Un nome breve, che è un nome tutto minuscolo, come `"simplephp"` quello generato OpsWorks dal nome dell'app (stringa).

```
node["opsworks"]["applications"]["i"]["slug_name"]
```

Attributi instance

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

L'attributo `instance` contiene una serie di attributi che specificano la configurazione di questa istanza.

architecture	availability_zone	backends
aws_instance_id	hostname	id
instance_type	ip	livelli
private_dns_name	private_ip	public_dns_name
Regione		

architecture

L'architettura dell'istanza, ad esempio `"i386"` (stringa).

```
node["opsworks"]["instance"]["architecture"]
```

availability_zone

La zona di disponibilità dell'istanza, ad esempio `"us-west-2a"` (stringa).

```
node["opsworks"]["instance"]["availability_zone"]
```

backends

Il numero di processi Web di back-end (stringa). Determina, ad esempio, il numero di connessioni simultanee che HAProxy verranno inoltrate a un backend Rails. Il valore predefinito dipende dal numero dei core e dalla memoria dell'istanza.

```
node["opsworks"]["instance"]["backends"]
```

aws_instance_id

L'ID dell' EC2 istanza (stringa).

```
node["opsworks"]["instance"]["aws_instance_id"]
```

hostname

Il nome host, ad esempio `"php-app1"` (stringa).

```
node["opsworks"]["instance"]["hostname"]
```

id

L'ID dell'istanza, che è un GUID AWS OpsWorks generato da Stacks che identifica in modo univoco l'istanza (stringa).

```
node["opsworks"]["instance"]["id"]
```

instance_type

Il tipo di istanza, ad esempio `"c1.medium"` (stringa).

```
node["opsworks"]["instance"]["instance_type"]
```

ip

L'indirizzo IP pubblico (stringa).

```
node["opsworks"]["instance"]["ip"]
```

livelli

Un elenco dei livelli dell'istanza, identificati dai relativi nomi brevi, ad esempio `"lb"` o `"db-master"` (elenco di stringhe).

```
node["opsworks"]["instance"]["layers"]
```

private_dns_name

Il nome DNS privato (stringa).

```
node["opsworks"]["instance"]["private_dns_name"]
```

private_ip

L'indirizzo IP privato (stringa).

```
node["opsworks"]["instance"]["private_ip"]
```

public_dns_name

Il nome DNS pubblico (stringa).

```
node["opsworks"]["instance"]["public_dns_name"]
```

Regione

La regione AWS, ad esempio `"us-west-2"` (stringa).

```
node["opsworks"]["instance"]["region"]
```

Attributi layers

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

L'attributo `layers` contiene un set di attributi di livello, uno per ciascun livello dello stack, che vengono denominati con il nome breve del livello, ad esempio `php-app`. Uno stack può avere al massimo uno dei livelli integrati, i cui nomi brevi sono i seguenti:

- `db-master`: livello MySQL
- `java-app`: livello Java App Server
- `lb`: HAProxy livello
- `monitoring-master`: Strato di gangli
- `memcached`: livello Memcached
- `nodejs-app`: livello Node.js App Server
- `php-app`: livello PHP App Server
- `rails-app`: livello Rails App Server
- `web`: livello statico del server Web

Uno stack può contenere un qualsiasi numero di livelli personalizzati, con nomi brevi definiti dall'utente.

Ogni attributo di livello include i seguenti attributi:

- id
- Istanze
- name

id

L'ID del livello, che è un GUID generato OpsWorks e che identifica in modo univoco il livello (stringa).

```
node["opsworks"]["layers"]["layershortname"]["id"]
```

Istanze

L'elemento `instances` contiene un set di attributi di istanza, uno per ogni istanza online del livello. Sono denominati con il nome host dell'istanza, ad esempio `php-app1`.

> **Note**
>
> L'elemento `instances` contiene solo le istanze che si trovano nello stato online quando vengono creati gli attributi di configurazione e distribuzione di un particolare stack.

Ogni elemento di istanza contiene i seguenti attributi:

availability_zone	aws_instance_id	backends
booted_at	created_at	elastic_ip
instance_type	ip	private_ip
public_dns_name	private_dns_name	Regione
status		

availability_zone

La zona di disponibilità, ad esempio `"us-west-2a"` (stringa).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["availability_zone"]
```

aws_instance_id

L'ID dell' EC2 istanza (stringa).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["aws_instance_id"]
```

backends

Il numero di processi Web di back-end (numero). Determina, ad esempio, il numero di connessioni simultanee che HAProxy verranno inoltrate a un backend Rails. Il valore predefinito dipende dal numero dei core e dalla memoria dell'istanza.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["backends"]
```

booted_at

L'ora in cui l' EC2 istanza è stata avviata, utilizzando il formato yyyy-mm-dddThh UTC:MM:SS +HH:mm (stringa). Ad esempio, `"2013-10-01T08:35:22+00:00"` corrisponde a 8:35:22 del 10 ott. 2013, senza differenza di fuso orario. Per ulteriori informazioni, consulta il protocollo ISO 8601.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["booted_at"]
```

created_at

L'ora in cui l' EC2 istanza è stata creata, utilizzando il formato UTC:MM:SS+HH:mm (stringa). yyyy-mm-dddThh Ad esempio, `"2013-10-01T08:35:22+00:00"` corrisponde a 8:35:22 del 10 ott. 2013, senza differenza di fuso orario. Per ulteriori informazioni, consulta il protocollo ISO 8601.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["created_at"]
```

elastic_ip

L'indirizzo IP elastico, che è impostato su null se non è disponibile nell'istanza (stringa).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["elastic_ip"]
```

instance_type

Il tipo di istanza, ad esempio `"c1.medium"` (stringa).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["instance_type"]
```

ip

L'indirizzo IP pubblico (stringa).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["ip"]
```

private_ip

L'indirizzo IP privato (stringa).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["private_ip"]
```

public_dns_name

Il nome DNS pubblico (stringa).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["public_dns_name"]
```

private_dns_name

Il nome DNS privato (stringa).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["private_dns_name"]
```

Regione

La regione AWS, ad esempio `"us-west-2"` (stringa).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["region"]
```

status

Lo stato (stringa). I valori possibili sono i seguenti:

- "requested"
- "booting"
- "running_setup"
- "online"
- "setup_failed"
- "start_failed"
- "terminating"
- "terminated"
- "stopped"
- "connection_lost"

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["status"]
```

nome

Il nome del livello, che viene utilizzato per rappresentare il livello nella console (stringa). Può essere definito dall'utente e non è necessariamente univoco.

```
node["opsworks"]["layers"]["layershortname"]["name"]
```

Attributi rails_stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

nome

Specifica lo stack rails ed è impostato su "apache_passenger" o "nginx_unicorn" (stringa).

```
node["opsworks"]["rails_stack"]["name"]
```

ricetta

La ricetta associata, che varia a seconda del fatto che utilizzi Passenger o Unicorn (stringa):

- Unicorn: `"unicorn::rails"`
- Passenger: `"passenger_apache2::rails"`

```
node["opsworks"]["rails_stack"]["recipe"]
```

restart_command

Il comando restart, che varia a seconda del fatto che utilizzi Passenger o Unicorn (stringa):

- Unicorn: `"../../shared/scripts/unicorn clean-restart"`
- Passenger: `"touch tmp/restart.txt"`

service

Il nome del servizio, che varia a seconda del fatto che utilizzi Passenger o Unicorn (stringa):

- Unicorn: `"unicorn"`
- Passenger: `"apache2"`

```
node["opsworks"]["rails_stack"]["service"]
```

Attributi stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Gli attributi `stack` specificano alcuni aspetti della configurazione dello stack, ad esempio le configurazioni del livello di servizio.

- elb-load-balancers
- id
- nome
- rds_instances
- vpc_id

elb-load-balancers

Contiene un elenco di oggetti incorporati, uno per ogni sistema di bilanciamento del carico di Elastic Load Balancing nello stack. Ogni oggetto incorporato contiene i seguenti attributi che descrivono la configurazione del sistema di bilanciamento del carico.

> **Note**
>
> La sintassi del nodo generale per questi attributi è riportata di seguito. *i* specifica l'indice di elenco a base zero dell'istanza.
>
> ```
> node["opsworks"]["stack"]["elb-load-balancers"]["i"]["attribute_name"]
> ```

dns_name

Il nome DNS del sistema di bilanciamento del carico (stringa).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["dns_name"]
```

nome

Il nome del sistema di bilanciamento del carico (stringa).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["name"]
```

layer_id

L'ID del livello a cui il sistema di bilanciamento del carico è collegato (stringa).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["layer_id"]
```

id

L'ID dello stack (stringa).

```
node["opsworks"]["stack"]["id"]
```

nome

Il nome dello stack (stringa).

```
node["opsworks"]["stack"]["name"]
```

rds_instances

Contiene un elenco di oggetti incorporati, uno per ogni istanza Amazon RDS registrata con lo stack. Ogni oggetto incorporato contiene un set di attributi che definiscono la configurazione dell'istanza. Puoi specificare questi valori quando utilizzi la console o l'API di Amazon RDS per creare l'istanza. Puoi anche utilizzare la console o l'API di Amazon RDS per modificare alcune impostazioni dopo la creazione dell'istanza. Per ulteriori informazioni, consulta la documentazione di Amazon RDS.

> **Note**
>
> La sintassi del nodo generale per questi attributi è riportata di seguito. *i* specifica l'indice di elenco a base zero dell'istanza.
>
> ```
> node["opsworks"]["stack"]["rds_instances"]["i"]["attribute_name"]
> ```

Se il tuo stack ha più istanze Amazon RDS, di seguito è riportato un esempio di come utilizzare una particolare istanza in una ricetta.

```
if my_rds = node["opsworks"]["stack"]["rds_instances"].select{|rds_instance|
 rds_instance["db_instance_identifier"] == ‘db_id’ }.first
  template “/etc/rds.conf” do
    source "rds.conf.erb"
    variables :address => my_rds["address"]
  end
```

```
end
```

address	allocated_storage	arn
auto_minor_version_upgrade	availability_zone	backup_retention_period
db_instance_class	db_instance_identifier	db_instance_status
db_name	db_parameter_groups	db_security_groups
db_user	engine	instance_create_time
license_model	multi_az	option_group_memberships
port	preferred_backup_window	preferred_maintenance_window
publicly_accessible	read_replica_db_instance_identifiers	Regione
status_infos	vpc_security_groups	

address

L'URL delle istanze, come `opsinstance.ccdvt3hwog1a.us-west-2.rds.amazonaws.com` (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["address"]
```

allocated_storage

Lo storage allocato, in GB (numero).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["allocated_storage"]
```

arn

L'ARN dell'istanza (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["arn"]
```

auto_minor_version_upgrade

Indica se l'aggiornamento a una versione secondaria deve essere eseguito automaticamente (booleano).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["auto_minor_version_upgrade"]
```

availability_zone

La zona di disponibilità dell'istanza, ad esempio `us-west-2a` (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["availability_zone"]
```

backup_retention_period

Il periodo di retention dei backup, in giorni (numero).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["backup_retention_period"]
```

db_instance_class

La classe dell'istanza di database, come `db.m1.small` (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_class"]
```

db_instance_identifier

L'identificatore istanze DB definito dall'utente (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_identifier"]
```

db_status

Lo stato dell'istanza (stringa). Per ulteriori informazioni, consulta la pagina relativa all'istanza database.

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_status"]
```

db_name

Il nome del database definito dall'utente (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_name"]
```

db_parameter_groups

I gruppi di parametri database dell'istanza, che contiene un elenco di oggetti incorporati, uno per ciascun gruppo di parametri. Per ulteriori informazioni, consulta Uso di gruppi di parametri database. Ogni oggetto contiene gli attributi seguenti:

db_parameter_group_name

Il nome del gruppo (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_parameter_groups"][j"]
["db_parameter_group_name"]
```

parameter_apply_status

Lo stato di applicazione (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_parameter_groups"][j"]
["parameter_apply_status"]
```

db_security_groups

I gruppi di sicurezza database dell'istanza, che contiene un elenco di oggetti incorporati, uno per ciascun gruppo di sicurezza. Per ulteriori informazioni, consulta Utilizzo dei gruppi di sicurezza database. Ogni oggetto contiene gli attributi seguenti

db_security_group_name

Il nome del gruppo di sicurezza (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_security_groups"][j"]
["db_security_group_name"]
```

status

Lo stato (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_security_groups"][j"]
["status"]
```

db_user

Il nome utente master definito dall'utente (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_user"]
```

engine

Il motore di database, ad esempio `mysql(5.6.13)` (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["engine"]
```

instance_create_time

L'ora di creazione dell'istanza, ad esempio `2014-04-15T16:13:34Z` (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["instance_create_time"]
```

license_model

Il modello di licenza dell'istanza, ad esempio `general-public-license` (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["license_model"]
```

multi_az

Indica se la implementazione Multi-AZ è abilitata (booleano).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["multi_az"]
```

option_group_memberships

Le appartenenze ai gruppi di opzioni dell'istanza, che contiene un elenco di oggetti incorporati, uno per ciascun gruppo di opzioni. Per ulteriori informazioni, consulta Uso di gruppi di opzioni. Ogni oggetto contiene gli attributi seguenti:

option_group_name

Il nome del gruppo (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["option_group_memberships"]
[j"]["option_group_name"]
```

status

Lo stato del gruppo (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["option_group_memberships"]
[j"]["status"]
```

port

La porta del server di database (numero).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["port"]
```

preferred_backup_window

La finestra di backup giornaliera preferita, ad esempio `06:26-06:56` (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["preferred_backup_window"]
```

preferred_maintenance_window

La finestra di manutenzione settimanale preferita, ad esempio `thu:07:13-thu:07:43` (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["preferred_maintenance_window"]
```

publicly_accessible

Indica se il database è accessibile pubblicamente (booleano).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["publicly_accessible"]
```

read_replica_db_instance_identifiers

Un elenco degli identificatori di istanza con replica di lettura (elenco di stringhe). Per ulteriori informazioni, consulta Uso di repliche di lettura.

```
node["opsworks"]["stack"]["rds_instances"]["i"]
["read_replica_db_instance_identifiers"]
```

Regione

La regione AWS, ad esempio `us-west-2` (stringa).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["region"]
```

status_infos

Un elenco di informazioni di stato (elenco di stringhe).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["status_infos"]
```

vpc_security_groups

Un elenco di gruppi di sicurezza VPC (elenco di stringhe).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["vpc_security_groups"]
```

vpc_id

L'ID VPC (stringa). Questo valore è `null` se l'istanza non è in un VPC.

```
node["opsworks"]["stack"]["vpc_id"]
```

Altri attributi opsworks di primo livello

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione include gli attributi `opsworks` che non dispongono di attributi figlio.

activity

L'attività associata agli attributi, ad esempio `deploy` (stringa).

```
node["opsworks"]["activity"]
```

agent_version

La versione dell' OpsWorks agente dell'istanza (stringa).

```
node["opsworks"]["agent_version"]
```

deploy_chef_provider

Il provider di distribuzione Chef, che influisce sulla struttura di directory dell'app distribuita (stringa). È possibile impostare questo attributo su uno dei seguenti:

- `Branch`
- `Revision`
- `Timestamped` (Valore predefinito)

```
node["opsworks"]["deploy_chef_provider"]
```

ruby_stack

Lo stack Ruby (stringa). L'impostazione predefinita è la versione enterprise (`ruby_enterprise`). Per la versione MRI, imposta questo attributo su `ruby`.

```
node["opsworks"]["ruby_stack"]
```

ruby_version

La versione Ruby che verrà utilizzata dalle applicazioni (stringa). Puoi usare questo attributo per specificare solo le versioni principale e secondaria. Devi utilizzare l'attributo ["ruby"] appropriato per specificare la versione della patch. Per ulteriori informazioni su come specificare una versione, inclusi gli esempi, consulta Versioni di Ruby. Per tutti i dettagli sul modo in cui AWS OpsWorks Stacks determina la versione di Ruby, consulta il file degli attributi integrati, ruby.rb.

```
node["opsworks"]["ruby_version"]
```

run_cookbook_tests

Se eseguire minitest-chef-handlerdei test sui libri di cucina di Chef 11.4 (booleano).

```
node["opsworks"]["run_cookbook_tests"]
```

sent_at

Indica quando questo comando è stato inviato all'istanza (numero).

```
node["opsworks"]["sent_at"]
```

implementazione

Se questi attributi sono associati a un'attività di distribuzione, `deployment` è impostato sull'ID della distribuzione, un GUID generato da AWS OpsWorks Stacks che identifica in modo univoco la distribuzione (stringa). In caso contrario, l'attributo è impostato su null.

```
node["opsworks"]["deployment"]
```

Attributi opsworks_custom_cookbooks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Contiene attributi che specificano i libri di ricette personalizzati dello stack.

abilitato

Indica se i libri di ricette personalizzati sono abilitati (booleano).

```
node["opsworks_custom_cookbooks"]["enabled"]
```

recipes

Un elenco delle ricette che devono essere eseguite per questo comando, incluse le ricette personalizzate, nel formato *cookbookname*::*recipename* (elenco di stringhe).

```
node["opsworks_custom_cookbooks"]["recipes"]
```

Attributi dependencies

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Contiene diversi attributi correlati al `update_dependencies`comando dello stackEsecuzione dei comandi dello stack.

gem_binary

La posizione del file binario Gems (stringa).

upgrade_debs

Indica se effettuare l'aggiornamento dei pacchetti Debs (booleano).

update_debs

Indica se effettuare l'aggiornamento dei pacchetti Debs (booleano).

Attributi ganglia

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Contiene un attributo web che include vari attributi che specificano come accedere alla pagina Web delle statistiche Ganglia:

password

La password necessaria per accedere alla pagina delle statistiche (stringa).

```
node["ganglia"]["web"]["password"]
```

url

Il percorso URL delle statistiche, ad esempio `"/ganglia"` (stringa). L'URL completo è `http://`*`DNSNameURLPath`*, dove *`DNSName`* è il nome DNS dell'istanza associata.

```
node["ganglia"]["web"]["url"]
```

Utente

Il nome utente necessario per accedere alla pagina delle statistiche (stringa).

```
node["ganglia"]["web"]["user"]
```

Attributi mysql

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Contiene un set di attributi che specificano la configurazione del server di database MySQL.

clients

Un elenco di indirizzi IP client (elenco di stringhe).

```
node["mysql"]["clients"]
```

server_root_password

La password root (stringa).

```
node["mysql"]["server_root_password"]
```

Attributi passenger

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Contiene un set di attributi che specificano la configurazione Phusion Passenger.

gem_bin

La posizione dei RubyGems file binari, ad esempio "`/usr/local/bin/gem`" (string).

```
node["passenger"]["gem_bin"]
```

max_pool_size

La dimensione massima del pool (numero).

```
node["passenger"]["max_pool_size"]
```

ruby_bin

La posizione dei file binari Ruby, ad esempio "`/usr/local/bin/ruby`".

```
node["passenger"]["ruby_bin"]
```

version

La versione Passenger (stringa).

```
node["passenger"]["version"]
```

Attributi opsworks_bundler

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Contiene elementi che specificano il supporto di Bundler.

manage_package

Indica se installare e gestire Bundler (booleano).

```
node["opsworks_bundler"]["manage_package"]
```

version

La versione di Bundler (stringa).

```
node["opsworks_bundler"]["version"]
```

Attributi deploy

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se gli attributi sono associati a un evento Deploy o a un comando dello stack Execute Recipes (Esegui ricette), l'attributo `deploy` contiene un attributo per ogni app che è stata distribuita, denominato con il nome breve dell'app. Ogni attributo di app contiene i seguenti attributi:

applicazione	application_type	auto_bundle_on_deploy
database	deploy_to	domains
document_root	environment_variables	gruppo
keep_releases	memcached	migrazione
mounted_at	purge_before_symlink	rails_env
restart_command	scm	ssl_certificate
ssl_certificate_ca	ssl_certificate_key	ssl_support
stack	symlink_before_migrate	symlinks
Utente		

applicazione

Il nome del campo dati dinamico dell'app, ad esempio `"simplephp"` (stringa).

```
node["deploy"]["appshortname"]["application"]
```

application_type

Il tipo di app (stringa). I valori possibili sono i seguenti:

- `java`: un'app Java
- `nodejs`: un'app Node.js
- `php`: un'app PHP
- `rails`: un'app Ruby on Rails
- `web`: una pagina HTML statica
- `other`: tutti gli altri tipi di applicazioni

```
node["deploy"]["appshortname"]["application_type"]
```

auto_bundle_on_deploy

Per applicazioni Rails, indica se eseguire il bundler durante la distribuzione (booleano).

```
node["deploy"]["appshortname"]["auto_bundle_on_deploy"]
```

database

Contiene le informazioni richieste per la connessione del database dell'app. Se all'app è collegato un livello di database, AWS OpsWorks Stacks assegna automaticamente i valori appropriati a questi attributi.

adapter

L'adattatore del database, ad esempio `mysql` (stringa).

```
node["deploy"]["appshortname"]["database"]["adapter"]
```

database

Il nome del database, che in genere è il nome del campo dati dinamico dell'app, ad esempio `"simplephp"` (stringa).

```
node["deploy"]["appshortname"]["database"]["database"]
```

data_source_provider

L'origine dati: `mysql` o `rds` (stringa).

```
node["deploy"]["appshortname"]["database"]["data_source_provider"]
```

host

L'indirizzo IP dell'host del database (stringa).

```
node["deploy"]["appshortname"]["database"]["host"]
```

password

La password del database (stringa).

```
node["deploy"]["appshortname"]["database"]["password"]
```

port

La porta del database (numero).

```
node["deploy"]["appshortname"]["database"]["port"]
```

reconnect

Per applicazioni Rails, indica se l'applicazione deve riconnettersi qualora la connessione non esista più (booleano).

```
node["deploy"]["appshortname"]["database"]["reconnect"]
```

username

Il nome utente (stringa).

```
node["deploy"]["appshortname"]["database"]["username"]
```

deploy_to

Indica dove deve essere distribuita l'app, ad esempio `"/srv/www/simplephp"` (stringa).

```
node["deploy"]["appshortname"]["deploy_to"]
```

domains

Un elenco dei domini dell'app (elenco di stringhe).

```
node["deploy"]["appshortname"]["domains"]
```

document_root

La radice del documento, se specifichi una radice non predefinita oppure null se utilizzi la radice predefinita (stringa).

```
node["deploy"]["appshortname"]["document_root"]
```

environment_variables

Una raccolta di non oltre venti attributi che rappresentano le variabili di ambiente specificate dall'utente e definite per l'app. Per ulteriori informazioni su come definire le variabili di ambiente di un'app, consulta Aggiunta di app. Ogni nome di attributo è impostato su un nome di variabile di ambiente e il valore corrispondente è impostato sul valore della variabile, pertanto puoi utilizzare la sintassi seguente per fare riferimento a un determinato valore.

```
node["deploy"]["appshortname"]["environment_variables"]["variable_name"]
```

gruppo

Il gruppo dell'app (stringa).

```
node["deploy"]["appshortname"]["group"]
```

keep_releases

Il numero di distribuzioni di app che AWS OpsWorks Stacks memorizzerà (numero). Questo attributo controlla il numero di volte che puoi eseguire il rollback di un'app. Per impostazione predefinita, è impostato sul valore globale, deploy_keep_releases , che ha un valore predefinito di 5. Puoi sostituire `keep_releases` per specificare il numero di distribuzioni archiviate per un'applicazione specifica.

```
node["deploy"]["appshortname"]["keep_releases"]
```

memcached

Contiene due attributi che definiscono la configurazione memcached.

host

L'indirizzo IP (stringa) dell'istanza del server Memcached.

```
node["deploy"]["appshortname"]["memcached"]["host"]
```

port

La porta sulla quale è in ascolto il server memcached (numero).

```
node["deploy"]["appshortname"]["memcached"]["port"]
```

migrazione

Per le applicazioni Rails, indica se eseguire migrazioni (booleano).

```
node["deploy"]["appshortname"]["migrate"]
```

mounted_at

Il punto di montaggio dell'app, se specifichi un punto di montaggio non predefinito oppure null se usi il punto di montaggio predefinito (stringa).

```
node["deploy"]["appshortname"]["mounted_at"]
```

purge_before_symlink

Per le app Rails, una vasta gamma di percorsi da cancellare prima di creare collegamenti simbolici (elenco di stringhe).

```
node["deploy"]["appshortname"]["purge_before_symlink"]
```

rails_env

Per le istanze di Rails App Server, l'ambiente rails, ad esempio (string). `"production"`

```
node["deploy"]["appshortname"]["rails_env"]
```

restart_command

Un comando da eseguire quando l'app viene riavviata, ad esempio `"echo 'restarting app'"`.

```
node["deploy"]["appshortname"]["restart_command"]
```

scm

Contiene un set di attributi che specificano le informazioni OpsWorks utilizzate per distribuire l'app dal relativo repository di controllo del codice sorgente. Gli attributi variano in base al tipo di repository.

password

La password per i repository privati e null per i repository pubblici (stringa). Per i bucket Amazon S3 privati, l'attributo è impostato sulla chiave segreta.

```
node["deploy"]["appshortname"]["scm"]["password"]
```

repository

L'URL del repository, come `"git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git"` (stringa).

```
node["deploy"]["appshortname"]["scm"]["repository"]
```

revisione

Se nel repository sono presenti più rami, l'attributo specifica il ramo o la versione dell'app, ad esempio `"version1"` (stringa). In alternativa, è impostato su null.

```
node["deploy"]["appshortname"]["scm"]["revision"]
```

scm_type

Il tipo di repository (stringa). I valori possibili sono i seguenti:

- `"git"`: un repository Git
- `"svn"`: un repository Subversion
- `"s3"`: Un bucket Amazon S3
- `"archive"`: un archivio HTTP
- `"other"`: un altro tipo di repository

```
node["deploy"]["appshortname"]["scm"]["scm_type"]
```

ssh_key

Una chiave SSH di distribuzione per l'accesso ai repository Git privati e null per i repository pubblici (stringa).

```
node["deploy"]["appshortname"]["scm"]["ssh_key"]
```

Utente

Il nome utente per i repository privati e null per i repository pubblici (stringa). Per i bucket Amazon S3 privati, l'attributo è impostato sulla chiave di accesso.

```
node["deploy"]["appshortname"]["scm"]["user"]
```

ssl_certificate

Il certificato SSL dell'app, se hai abilitato il supporto SSL o, altrimenti, null (stringa).

```
node["deploy"]["appshortname"]["ssl_certificate"]
```

ssl_certificate_ca

Se SSL è abilitato, un attributo per specificare una chiave di autorità di certificazione intermedia o l'autenticazione client (stringa).

```
node["deploy"]["appshortname"]["ssl_certificate_ca"]
```

ssl_certificate_key

La chiave privata SSL dell'app, se hai abilitato il supporto SSL o, altrimenti, null (stringa).

```
node["deploy"]["appshortname"]["ssl_certificate_key"]
```

ssl_support

Indica se SSL è supportato (booleano).

```
node["deploy"]["appshortname"]["ssl_support"]
```

stack

Contiene un attributo booleano, `needs_reload`, che specifica se ricaricare il server app durante la distribuzione.

```
node["deploy"]["appshortname"]["stack"]["needs_reload"]
```

symlink_before_migrate

Per le app Rails, contiene collegamenti simbolici che devono essere creati prima di eseguire migrazioni come coppie `"link":"target"`.

```
node["deploy"]["appshortname"]["symlink_before_migrate"]
```

symlinks

Contiene i collegamenti simbolici della distribuzione come coppie `"link":"target"`.

```
node["deploy"]["appshortname"]["symlinks"]
```

Utente

L'utente dell'app (stringa).

```
node["deploy"]["appshortname"]["user"]
```

Altri attributi di primo livello

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione include attributi di configurazione dello stack di primo livello che non dispongono di attributi figlio.

Attributi rails

Contiene un attributo max_pool_size che specifica la dimensione massima del pool del server (numero). Il valore dell'attributo è impostato da AWS OpsWorks Stacks e dipende dal tipo di istanza, ma puoi sovrascriverlo utilizzando un JSON personalizzato o un file di attributi personalizzato.

```
node["rails"]["max_pool_size"]
```

Attributi recipes

Un elenco delle ricette integrate eseguite da questa attività, nel formato "*cookbookname*::*recipename*" (elenco di stringhe).

```
node["recipes"]
```

Attributi opsworks_rubygems

Contiene un elemento di versione che specifica la RubyGems versione (stringa).

```
node["opsworks_rubygems"]["version"]
```

Attributi languages

Contiene un attributo per ogni linguaggio installato, denominato in base al linguaggio, ad esempio ruby. L'attributo è un oggetto che contiene un attributo, ad esempio ruby_bin, che specifica la cartella di installazione, ad esempio "`/usr/bin/ruby`" (stringa).

Attributi ssh_users

Contiene un set di attributi, ognuno dei quali descrive uno degli utenti a cui sono state concesse le autorizzazioni SSH. Ogni attributo è denominato con l'ID Unix di un utente. AWS OpsWorks Stacks genera un ID univoco per ogni utente nell'intervallo 2000-4000, ad esempio"`2001`", e crea un utente con quell'ID su ogni istanza. Poiché AWS OpsWorks riserva l'intervallo 2000-4000, gli utenti che crei al di fuori AWS OpsWorks (utilizzando ricette di libri di cucina o importando utenti da IAM, ad esempio) possono far sì UIDs che questi vengano sovrascritti AWS OpsWorks da Stacks per un altro utente. AWS OpsWorks Come best practice, crea utenti e gestisci il loro accesso nella console Stacks. AWS OpsWorks Se crei utenti al di fuori di AWS OpsWorks Stacks, utilizza *UnixID* valori superiori a 4000.

Ogni attributo contiene i seguenti attributi:

e-mail

L'indirizzo e-mail dell'utente (stringa).

```
node["ssh_users"]["UnixID"]["email"]
```

public_key

La chiave SSH pubblica dell'utente (stringa).

```
node["ssh_users"]["UnixID"]["public_key"]
```

sudoer

Indica se l'utente dispone di autorizzazioni sudo (booleano).

```
node["ssh_users"]["UnixID"]["sudoer"]
```

nome

Il nome utente (stringa).

```
node["ssh_users"]["UnixID"]["name"]
```

Attributi dei libri di ricette predefiniti

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Note

La maggior parte di questi attributi è disponibile solo per gli stack Linux.

La maggior parte delle ricette predefinite dispone di uno o più file di attributi che definiscono varie impostazioni. Puoi accedere a queste impostazioni nelle ricette personalizzate e utilizzare dati JSON personalizzati per sostituirle. In genere è necessario accedere o sovrascrivere gli attributi che controllano la configurazione delle varie tecnologie server supportate da Stacks. AWS OpsWorks In questa sezione sono riepilogati questi attributi. I file completi degli attributi e le ricette e i modelli associati sono disponibili all'indirizzo https://github.com/aws/ opsworks-cookbooks.git.

> **Note**
>
> Tutti gli attributi di ricetta predefiniti sono di tipo `default`.

Argomenti

- Attributi apache2
- Attributi deploy
- Attributi haproxy
- Attributi memcached
- Attributi mysql
- Attributi nginx
- Attributi opsworks_berkshelf
- Attributi opsworks_java
- Attributi passenger_apache2
- Attributi ruby
- Attributi unicorn

Attributi apache2

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi attributi sono disponibili solo per gli stack Linux.

Gli attributi apache2 specificano la configurazione del server HTTP Apache. Per ulteriori dettagli, consulta l'argomento relativo alle caratteristiche d base di Apache. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

binary	contact	deflate_types
dir	document_root	gruppo
hide_info_headers	icondir	init_script
keepalive	keepaliverequests	keepalivetimeout
lib_dir	libexecdir	listen_ports
log_dir	Attributi logrotate	pid_file
Attributi prefork	serversignature	servertokens
timeout	traceenable	Utente
version	Attributi worker	

binary

Posizione del file binario Apache (stringa). Il valore predefinito è `'/usr/sbin/httpd'`.

```
node[:apache][:binary]
```

contact

Contatto e-mail (stringa). Il valore predefinito è l'indirizzo fittizio `'ops@example.com'`.

```
node[:apache][:contact]
```

deflate_types

Indica a `mod_deflate` di abilitare la compressione per i tipi Mime specificati, se sono supportati dal browser (elenco di stringhe). Il valore predefinito è riportato di seguito:

```
['application/javascript',
 'application/json',
 'application/x-javascript',
 'application/xhtml+xml',
 'application/xml',
 'application/xml+rss',
 'text/css',
 'text/html',
 'text/javascript',
 'text/plain',
 'text/xml']
```

> **Warning**
>
> La compressione può introdurre rischi a livello di sicurezza. Per disabilitare completamente la compressione, imposta questo attributo come segue:
>
> ```
> node[:apache][:deflate_types] = []
> ```

```
node[:apache][:deflate_types]
```

dir

Directory radice del server (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e Red Hat Enterprise Linux (RHEL): `'/etc/httpd'`
- Ubuntu: `'/etc/apache2'`

```
node[:apache][:dir]
```

document_root

Radice del documento (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `'/var/www/html'`
- Ubuntu: `'/var/www'`

```
node[:apache][:document_root]
```

gruppo

Il nome del gruppo (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `'apache'`
- Ubuntu: `'www-data'`

```
node[:apache][:group]
```

hide_info_headers

Specifica se omettere le informazioni su versione e modulo nelle intestazioni HTTP (`'true'`/`'false'`) (stringa). Il valore predefinito è `'true'`.

```
node[:apache][:hide_info_headers]
```

icondir

Directory delle icone (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `'/var/www/icons/'`
- Ubuntu: `'/usr/share/apache2/icons'`

```
node[:apache][:icondir]
```

init_script

Script di inizializzazione (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `'/etc/init.d/httpd'`
- Ubuntu: `'/etc/init.d/apache2'`

```
node[:apache][:init_script]
```

keepalive

Indica se abilitare le connessioni keep-alive (stringa). I valori possibili sono `'On'` e `'Off'` (stringa). Il valore predefinito è `'Off'`.

```
node[:apache][:keepalive]
```

keepaliverequests

Numero massimo di richieste keep-alive che Apache è in grado di gestire contemporaneamente (numero). Il valore predefinito è `100`.

```
node[:apache][:keepaliverequests]
```

keepalivetimeout

Intervallo di tempo durante il quale Apache rimane in attesa di una richiesta prima di chiudere la connessione (numero). Il valore predefinito è `3`.

```
node[:apache][:keepalivetimeout]
```

lib_dir

Directory contenente le librerie del codice degli oggetti (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux (x86): `'/usr/lib/httpd'`
- Amazon Linux (x64) e RHEL: `'/usr/lib64/httpd'`
- Ubuntu: `'/usr/lib/apache2'`

```
node[:apache][:lib_dir]
```

libexecdir

Directory contenente gli eseguibili dei programmi (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux (x86): `'/usr/lib/httpd/modules'`
- Amazon Linux (x64) e RHEL: `'/usr/lib64/httpd/modules'`

- Ubuntu: `'/usr/lib/apache2/modules'`

```
node[:apache][:libexecdir]
```

listen_ports

Elenco di porte che il server ascolta (elenco di stringhe). Il valore predefinito è `[ '80','443' ]`.

```
node[:apache][:listen_ports]
```

log_dir

Directory dei log (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `'/var/log/httpd'`
- Ubuntu: `'/var/log/apache2'`

```
node[:apache][:log_dir]
```

Attributi logrotate

Questi attributi specificano la modalità di rotazione dei file di log.

delaycompress

Indica se ritardare la compressione di un file di log chiuso fino all'inizio del successivo ciclo di rotazione (`'true'`/`'false'`) (stringa). Il valore predefinito è `'true'`.

```
node[:apache][:logrotate][:delaycompress]
```

gruppo

Gruppo di file di log (stringa). Il valore predefinito è `'adm'`.

```
node[:apache][:logrotate][:group]
```

mode

Modalità dei file di log (stringa). Il valore predefinito è `'640'`.

```
node[:apache][:logrotate][:mode]
```

owner

Proprietario dei file di log (stringa). Il valore predefinito è `'root'`.

```
node[:apache][:logrotate][:owner]
```

rotate

Numero di cicli di rotazione prima della rimozione di un file di log chiuso (stringa). Il valore predefinito è `'30'`.

```
node[:apache][:logrotate][:rotate]
```

schedule

Pianificazione della rotazione (stringa). I valori possibili sono i seguenti:

- `'daily'`
- `'weekly'`
- `'monthly'`

Il valore predefinito è `'daily'`.

```
node[:apache][:logrotate][:schedule]
```

pid_file

File contenente l'ID processo del daemon (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `'/var/run/httpd/httpd.pid'`
- Ubuntu: `'/var/run/apache2.pid'`

```
node[:apache][:pid_file]
```

Attributi prefork

Questi attributi specificano la configurazione delle operazioni precedenti all'esecuzione del fork.

maxclients

Numero massimo di richieste simultanee che verranno servite (numero). Il valore predefinito è `400`.

> **Note**
>
> Usa questo attributo solo per le istanze che eseguono Amazon Linux o RHEL. Se le istanze eseguono Ubuntu 14.04 LTS, utilizzare maxrequestworkers.

```
node[:apache][:prefork][:maxclients]
```

maxrequestsperchild

Numero massimo di richieste che un processo server figlio gestirà (numero). Il valore predefinito è `10000`.

```
node[:apache][:prefork][:maxrequestsperchild]
```

maxrequestworkers

Numero massimo di richieste simultanee che verranno servite (numero). Il valore predefinito è `400`.

> **Note**
>
> Utilizzare questo attributo solo per le istanze che eseguono Ubuntu 14.04 LTS. Se le tue istanze eseguono Amazon Linux o RHEL, usa. maxclients

```
node[:apache][:prefork][:maxrequestworkers]
```

maxspareservers

Numero massimo di processi server figlio inattivi (numero). Il valore predefinito è 32.

```
node[:apache][:prefork][:maxspareservers]
```

minspareservers

Numero minimo di processi server figlio inattivi (numero). Il valore predefinito è 16.

```
node[:apache][:prefork][:minspareservers]
```

serverlimit

Numero massimo di processi che possono essere configurati (numero). Il valore predefinito è `400`.

```
node[:apache][:prefork][:serverlimit]
```

startservers

Numero di processi server figlio da creare all'avvio (numero). Il valore predefinito è 16.

```
node[:apache][:prefork][:startservers]
```

serversignature

Specifica se e come configurare un piè di pagina finale per i documenti generati dal server (stringa). I valori possibili sono `'On'`, `'Off'` e `'Email'`). Il valore predefinito è `'Off'`.

```
node[:apache][:serversignature]
```

servertokens

Specifica il tipo di informazioni sulla versione del server incluse nell'intestazione della risposta (stringa):

- `'Full'`: informazioni complete. Ad esempio, Server: Apache/2.4.2 (Unix) PHP/4.2.2 /1.2 MyMod
- `'Prod'`: nome del prodotto. Ad esempio, Server: Apache
- `'Major'`: versione principale. Ad esempio, Server: Apache/2
- `'Minor'`: versione principale e versione secondaria. Ad esempio, Server: Apache/2.4
- `'Min'`: versione minima. Ad esempio, Server: Apache/2.4.2
- `'OS'`: versione con sistema operativo. Ad esempio, Server: Apache/2.4.2 (Unix)

Il valore predefinito è `'Prod'`.

```
node[:apache][:servertokens]
```

timeout

La quantità di tempo di attesa di Apache (numero). I/O Il valore predefinito è `120`.

```
node[:apache][:timeout]
```

traceenable

Indica se abilitare le richieste TRACE (stringa). I valori possibili sono `'On'` e `'Off'`. Il valore predefinito è `'Off'`.

```
node[:apache][:traceenable]
```

Utente

Il nome utente (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `'apache'`
- Ubuntu: `'www-data'`

```
node[:apache][:user]
```

version

Versione di Apache (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux: `2.2`
- Ubuntu 14.04 LTS: `2.4`
- RHEL: `2.4`

```
node[:apache][:version]
```

Attributi worker

Questi attributi specificano la configurazione dei processi di lavoro.

startservers

Numero di processi server figlio da creare all'avvio (numero). Il valore predefinito è `4`.

```
node[:apache][:worker][:startservers]
```

maxclients

Numero massimo di richieste simultanee che verranno servite (numero). Il valore predefinito è `1024`.

```
node[:apache][:worker][:maxclients]
```

maxsparethreads

Numero massimo di thread inattivi (numero). Il valore predefinito è 192.

```
node[:apache][:worker][:maxsparethreads]
```

minsparethreads

Numero minimo di thread inattivi (numero). Il valore predefinito è 64.

```
node[:apache][:worker][:minsparethreads]
```

threadsperchild

Numero di thread per il processo figlio (numero). Il valore predefinito è 64.

```
node[:apache][:worker][:threadsperchild]
```

maxrequestsperchild

Numero massimo di richieste che un processo server figlio gestirà (numero). Il valore predefinito è `10000`.

```
node[:apache][:worker][:maxrequestsperchild]
```

Attributi deploy

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Il file di attributi `deploy.rb` del libro di ricette di ditribuzione predefinito definisce i seguenti attributi nello spazio dei nomi `opsworks`. Per ulteriori informazioni sulle directory di distribuzione, consulta Ricette di ditribuzione. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

deploy_keep_releases

Un'impostazione globale per il numero di distribuzioni di app che AWS OpsWorks Stacks memorizzerà (numero). Il valore predefinito è 5. Questo valore controlla il numero di volte che puoi eseguire il rollback di un'app.

```
node[:opsworks][:deploy_keep_releases]
```

gruppo

(Solo Linux) Impostazione `group` per la directory di distribuzione dell'app (stringa). Il valore predefinito dipende dal sistema operativo dell'istanza:

- Per le istanze Ubuntu, il valore predefinito è `www-data`.
- Per le istanze Amazon Linux o RHEL che sono membri di un livello Rails App Server che utilizza Nginx e Unicorn, il valore predefinito è. `nginx`
- Per tutte le altre istanze Amazon Linux o RHEL, il valore predefinito è `apache`.

```
node[:opsworks][:deploy_user][:group]
```

Utente

(Solo Linux) Impostazione `user` per la directory di distribuzione dell'app (stringa). Il valore predefinito è `deploy`.

```
node[:opsworks][:deploy_user][:user]
```

Attributi haproxy

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi attributi sono disponibili solo per gli stack Linux.

Gli `haproxy`attributi specificano la configurazione del HAProxy server. Per ulteriori informazioni, consulta HAProxyDocumenti. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

balance	check_interval	client_timeout
connect_timeout	default_max_connections	global_max_connections
health_check_method	health_check_url	queue_timeout
http_request_timeout	maxcon_factor_nodejs_app	maxcon_factor_nodejs_app_ssl
maxcon_factor_php_app	maxcon_factor_php_app_ssl	maxcon_factor_rails_app
maxcon_factor_rails_app_ssl	maxcon_factor_static	maxcon_factor_static_ssl
retries	server_timeout	stats_url
stats_user		

balance

Algoritmo utilizzato da un sistema di bilanciamento del carico per selezionare un server (stringa). Il valore predefinito è `'roundrobin'`. Le altre opzioni sono:

- 'static-rr'
- 'leastconn'
- 'source'

- 'uri'
- 'url_param'
- 'hdr(name)'
- 'rdp-cookie'
- 'rdp-cookie(name)'

Per ulteriori informazioni su questi argomenti, consulta balance.

```
node[:haproxy][:balance]
```

check_interval

Intervallo di tempo del controllo dello stato (stringa). Il valore predefinito è `'10s'`.

```
node[:haproxy][:check_interval]
```

client_timeout

Durata massima dell'inattività di un client (stringa). Il valore predefinito è `'60s'`.

```
node[:haproxy][:client_timeout]
```

connect_timeout

Il tempo massimo di attesa per il HAProxy successo di un tentativo di connessione al server (stringa). Il valore predefinito è `'10s'`.

```
node[:haproxy][:connect_timeout]
```

default_max_connections

Numero massimo predefinito di connessioni (stringa). Il valore predefinito è `'80000'`.

```
node[:haproxy][:default_max_connections]
```

global_max_connections

Numero massimo di connessioni (stringa). Il valore predefinito è `'80000'`.

```
node[:haproxy][:global_max_connections]
```

health_check_method

Metodo di controllo dello stato (stringa). Il valore predefinito è `'OPTIONS'`.

```
node[:haproxy][:health_check_method]
```

health_check_url

Percorso URL utilizzato per controllare lo stato dei server (stringa). Il valore predefinito è `'/'`.

```
node[:haproxy][:health_check_url ]
```

queue_timeout

Tempo massimo per l'attesa di una connessione libera (stringa). Il valore predefinito è `'120s'`.

```
node[:haproxy][:queue_timeout]
```

http_request_timeout

Il tempo massimo di attesa per una richiesta HTTP completa (stringa). HAProxy Il valore predefinito è `'30s'`.

```
node[:haproxy][:http_request_timeout]
```

retries

Numero di tentativi di connessione dopo un errore di connessione al server (stringa). Il valore predefinito è `'3'`.

```
node[:haproxy][:retries]
```

server_timeout

Durata massima dell'inattività di un client (stringa). Il valore predefinito è `'60s'`.

```
node[:haproxy][:server_timeout]
```

stats_url

Percorso URL della pagina delle statistiche (stringa). Il valore predefinito è `'/haproxy?stats'`.

```
node[:haproxy][:stats_url]
```

stats_user

Nome utente della pagina relativa alle statistiche (stringa). Il valore predefinito è `'opsworks'`.

```
node[:haproxy][:stats_user]
```

Gli `maxcon` attributi rappresentano un moltiplicatore del fattore di carico utilizzato per calcolare il numero massimo di connessioni HAProxy consentite per i backend. Ad esempio, supponiamo di avere un server di app Rails su una piccola istanza con un `backend` valore di 4, il che significa che AWS OpsWorks Stacks configurerà quattro processi Rails per quell'istanza. Se si utilizza il `maxcon_factor_rails_app` valore predefinito 7, HAProxy gestirà 28 (4*7) connessioni al server Rails.

maxcon_factor_nodejs_app

Fattore maxcon per un server di applicazioni Node.js (numero). Il valore predefinito è `10`.

```
node[:haproxy][:maxcon_factor_nodejs_app]
```

maxcon_factor_nodejs_app_ssl

Fattore maxcon per un server di applicazioni Node.js con SSL (numero). Il valore predefinito è `10`.

```
node[:haproxy][:maxcon_factor_nodejs_app_ssl]
```

maxcon_factor_php_app

Fattore maxcon per un server di applicazioni PHP (numero). Il valore predefinito è `10`.

```
node[:haproxy][:maxcon_factor_php_app]
```

maxcon_factor_php_app_ssl

Fattore maxcon per un server di applicazioni PHP con SSL (numero). Il valore predefinito è `10`.

```
node[:haproxy][:maxcon_factor_php_app_ssl]
```

maxcon_factor_rails_app

Fattore maxcon per un server di applicazioni Rails (numero). Il valore predefinito è 7.

```
node[:haproxy][:maxcon_factor_rails_app]
```

maxcon_factor_rails_app_ssl

Fattore maxcon per un server di applicazioni Rails con SSL (numero). Il valore predefinito è 7.

```
node[:haproxy][:maxcon_factor_rails_app_ssl]
```

maxcon_factor_static

Fattore maxcon per un server Web statico (numero). Il valore predefinito è 15.

```
node[:haproxy][:maxcon_factor_static]
```

maxcon_factor_static_ssl

Fattore maxcon per un server Web statico con SSL (numero). Il valore predefinito è 15.

```
node[:haproxy][:maxcon_factor_static_ssl]
```

Attributi memcached

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Note

Questi attributi sono disponibili solo per gli stack Linux.

Gli attributi `memcached` specificano la configurazione del server Memcached. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

memory	max_connections	pid_file
port	start_command	stop_command
Utente		

memory

Quantità massima di memoria da utilizzare, in MB (numero). Il valore predefinito è `512`.

```
node[:memcached][:memory]
```

max_connections

Numero massimo di connessioni (stringa). Il valore predefinito è `'4096'`.

```
node[:memcached][:max_connections]
```

pid_file

File contenente l'ID processo del daemon (stringa). Il valore predefinito è `'var/run/memcached.pid'`.

```
node[:memcached][:pid_file]
```

port

Porta su cui rimanere in ascolto (numero). Il valore predefinito è `11211`.

```
node[:memcached][:port]
```

start_command

Comando di avvio (stringa). Il valore predefinito è `'/etc/init.d/memcached start'`.

```
node[:memcached][:start_command]
```

stop_command

Comando di arresto (stringa). Il valore predefinito è `'/etc/init.d/memcached stop'`.

```
node[:memcached][:stop_command]
```

Utente

Utente (stringa). Il valore predefinito è `'nobody'`.

```
node[:memcached][:user]
```

Attributi mysql

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi attributi sono disponibili solo per gli stack Linux.

Gli attributi `mysql` specificano la configurazione del master MySQL. Per ulteriori informazioni, consulta l'argomento relativo alle variabili di sistema del server. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

basedir	bind_address	clients

conf_dir	confd_dir	datadir
grants_path	mysql_bin	mysqladmin_bin
pid_file	port	root_group
server_root_password	socket	Attributi tunable

basedir

Directory di base (stringa). Il valore predefinito è `'/usr'`.

```
node[:mysql][:basedir]
```

bind_address

Indirizzo su cui MySQL rimane in ascolto (stringa). Il valore predefinito è `'0.0.0.0'`.

```
node[:mysql][:bind_address]
```

clients

Elenco di client (elenco di stringhe).

```
node[:mysql][:clients]
```

conf_dir

Directory contenente il file di configurazione (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `'/etc'`
- Ubuntu: `'/etc/mysql'`

```
node[:mysql][:conf_dir]
```

confd_dir

Directory contenente i file di configurazione aggiuntivi (stringa). Il valore predefinito è `'/etc/mysql/conf.d'`.

```
node[:mysql][:confd_dir]
```

datadir

Directory di dati (stringa). Il valore predefinito è `'/var/lib/mysql'`.

```
node[:mysql][:datadir]
```

grants_path

Posizione delle tabelle delle autorizzazioni (stringa). Il valore predefinito è `'/etc/mysql_grants.sql'`.

```
node[:mysql][:grants_path]
```

mysql_bin

Posizione dei file binari mysql (stringa). Il valore predefinito è `'/usr/bin/mysql'`.

```
node[:mysql][:mysql_bin]
```

mysqladmin_bin

Posizione di mysqladmin (stringa). Il valore predefinito è `'/usr/bin/mysqladmin'`.

```
node[:mysql][:mysqladmin_bin]
```

pid_file

File contenente l'ID processo del daemon (stringa). Il valore predefinito è `'/var/run/mysqld/mysqld.pid'`.

```
node[:mysql][:pid_file]
```

port

Porta su cui il server rimane in ascolto (numero). Il valore predefinito è `3306`.

```
node[:mysql][:port]
```

root_group

Gruppo radice (stringa). Il valore predefinito è `'root'`.

```
node[:mysql][:root_group]
```

server_root_password

Password radice del server (stringa). Il valore predefinito viene generato casualmente.

```
node[:mysql][:server_root_password]
```

socket

Posizione del file socket (stringa). Il valore predefinito è `'/var/lib/mysql/mysql.sock'`. I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `'/var/lib/mysql/mysql.sock'`
- Ubuntu: `'/var/run/mysqld/mysqld.sock'`

```
node[:mysql][:socket]
```

Attributi tunable

Gli attributi tunable vengono utilizzati per il tuning delle prestazioni.

back_log	innodb_additional_mem_pool_size	innodb_buffer_pool_size
innodb_flush_log_at_trx_commit	innodb_lock_wait_timeout	key_buffer
log_slow_queries	long_query_time	max_allowed_packet
max_connections	max_heap_table_size	net_read_timeout
net_write_timeout	query_cache_limit	query_cache_size
query_cache_type	thread_cache_size	thread_stack
wait_timeout	table_cache	

back_log

Numero massimo di richieste in sospeso (stringa). Il valore predefinito è `'128'`.

```
node[:mysql][:tunable][:back_log]
```

innodb_additional_mem_pool_size

Dimensione del pool utilizzato da Innodb per archiviare le strutture di dati interne (stringa). Il valore predefinito è `'20M'`.

```
node[:mysql][:tunable][:innodb_additional_mem_pool_size]
```

innodb_buffer_pool_size

Dimensione del pool di buffer Innodb (stringa). Il valore dell'attributo è impostato da AWS OpsWorks Stacks e dipende dal tipo di istanza, ma puoi sovrascriverlo utilizzando un JSON personalizzato o un file di attributi personalizzato.

```
node[:mysql][:tunable][:innodb_buffer_pool_size]
```

innodb_flush_log_at_trx_commit

Specifica la frequenza con cui Innodb svuota il buffer dei log (stringa). Il valore predefinito è `'2'`. Per ulteriori informazioni, consulta innodb_flush_log_at_trx_commit.

```
node[:mysql][:tunable][:innodb_flush_log_at_trx_commit]
```

innodb_lock_wait_timeout

Intervallo di tempo massimo, in secondi, durante il quale una transazione Innodb rimane in attesa di un blocco di riga (stringa). Il valore predefinito è `'50'`.

```
node[:mysql][:tunable][:innodb_lock_wait_timeout]
```

key_buffer

Dimensione del buffer degli indici (stringa). Il valore predefinito è `'250M'`.

```
node[:mysql][:tunable][:key_buffer]
```

log_slow_queries

Posizione del file di log delle query con tempi di risposta molto lunghi (stringa). Il valore predefinito è `'/var/log/mysql/mysql-slow.log'`.

```
node[:mysql][:tunable][:log_slow_queries]
```

long_query_time

Intervallo di tempo, in secondi, necessario per definire un query come query con tempi di risposta lunghi (stringa). Il valore predefinito è `'1'`.

```
node[:mysql][:tunable][:long_query_time]
```

max_allowed_packet

Dimensione di pacchetto massima consentita (stringa). Il valore predefinito è `'32M'`.

```
node[:mysql][:tunable][:max_allowed_packet]
```

max_connections

Numero massimo di connessioni client simultanee (stringa). Il valore predefinito è `'2048'`.

```
node[:mysql][:tunable][:max_connections]
```

max_heap_table_size

Dimensioni massime delle tabelle `MEMORY` create dall'utente (stringa). Il valore predefinito è `'32M'`.

```
node[:mysql][:tunable][:max_heap_table_size]
```

net_read_timeout

Intervallo di tempo, in secondi, durante il quale attendere altri dati da una connessione (stringa). Il valore predefinito è `'30'`.

```
node[:mysql][:tunable][:net_read_timeout]
```

net_write_timeout

Intervallo di tempo, in secondi, durante il quale attendere la scrittura di un blocco di dati in una connessione (stringa). Il valore predefinito è `'30'`.

```
node[:mysql][:tunable][:net_write_timeout]
```

query_cache_limit

Dimensioni massime di una singola query memorizzata nella cache (stringa). Il valore predefinito è `'2M'`.

```
node[:mysql][:tunable][:query_cache_limit]
```

query_cache_size

Dimensioni della cache di query (stringa). Il valore predefinito è `'128M'`.

```
node[:mysql][:tunable][:query_cache_size]
```

query_cache_type

Tipo di cache di query (stringa). I valori possibili sono i seguenti:

- `'0'`: nessun caching o nessun recupero di dati memorizzati nella cache.
- `'1'`: istruzioni di caching che non iniziano con `SELECT SQL_NO_CACHE`.
- `'2'`: istruzioni di caching che iniziano con `SELECT SQL_CACHE`.

Il valore predefinito è `'1'`.

```
node[:mysql][:tunable][:query_cache_type]
```

thread_cache_size

Numero di thread client che vengono memorizzati nella cache per il riutilizzo (stringa). Il valore predefinito è `'8'`.

```
node[:mysql][:tunable][:thread_cache_size]
```

thread_stack

Dimensione dello stack per ogni thread (stringa). Il valore predefinito è `'192K'`.

```
node[:mysql][:tunable][:thread_stack]
```

wait_timeout

Intervallo di tempo, in secondi, durante il quale attendere una connessione non interattiva. Il valore di default è `'180'` (stringa).

```
node[:mysql][:tunable][:wait_timeout]
```

table_cache

Numero di tabelle aperte (stringa). Il valore predefinito è `'2048'`.

```
node[:mysql][:tunable][:table_cache]
```

Attributi nginx

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Note

Questi attributi sono disponibili solo per gli stack Linux.

Gli attributi `nginx` specificano la configurazione di Nginx. Per ulteriori informazioni, consulta l'indice delle direttive. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

binary	dir	gzip

gzip_comp_level	gzip_disable	gzip_http_version
gzip_proxied	gzip_static	gzip_types
gzip_vary	keepalive	keepalive_timeout
log_dir	Utente	server_names_hash_bucket_size
worker_processes	worker_connections	

binary

Posizione dei file binari Nginx (stringa). Il valore predefinito è `'/usr/sbin/nginx'`.

```
node[:nginx][:binary]
```

dir

Posizione dei file quali, ad esempio, i file di configurazione (stringa). Il valore predefinito è `'/etc/nginx'`.

```
node[:nginx][:dir]
```

gzip

Indica se la compressione gzip è abilitata (stringa). I valori possibili sono `'on'` e `'off'`. Il valore predefinito è `'on'`.

> **Warning**
>
> La compressione può introdurre rischi a livello di sicurezza. Per disabilitare completamente la compressione, imposta questo attributo come segue:
>
> ```
> node[:nginx][:gzip] = 'off'
> ```

```
node[:nginx][:gzip]
```

gzip_comp_level

Il livello di compressione, che può variare da 1 a 9, dove 1 corrisponde alla compressione minima (stringa). Il valore predefinito è `'2'`.

```
node[:nginx][:gzip_comp_level]
```

gzip_disable

Disabilita la compressione gzip per gli agenti utente specificati (stringa). Il valore è un'espressione regolare e il valore predefinito è `'MSIE [1-6].(?!.*SV1)'`.

```
node[:nginx][:gzip_disable]
```

gzip_http_version

Abilita la compressione gzip per una versione HTTP specificata (stringa). Il valore predefinito è `'1.0'`.

```
node[:nginx][:gzip_http_version]
```

gzip_proxied

Indica se e come comprimere la risposta alle richieste proxy. I valori possibili (stringa) sono elencati di seguito:

- `'off'`: non comprime le richieste inviate tramite proxy
- `'expired'`: esegue la compressione se l'intestazione Expire impedisce il caching
- `'no-cache'`: esegue la compressione se l'intestazione Cache-Control è impostata su "no-cache"
- `'no-store'`: esegue la compressione se l'intestazione Cache-Control è impostata su "no-store"
- `'private'`: esegue la compressione se l'intestazione Cache-Control è impostata su "private"
- `'no_last_modified'`: esegue la compressione se l'attributo Last-Modified non è impostato
- `'no_etag'`: comprime se nella richiesta manca un'intestazione ETag
- `'auth'`: esegue la compressione se la richiesta include un'intestazione Authorization
- `'any'`: esegue la compressione di tutte le richieste inviate tramite proxy

Il valore predefinito è `'any'`.

```
node[:nginx][:gzip_proxied]
```

gzip_static

Indica se il modulo statico gzip è abilitato (stringa). I valori possibili sono `'on'` e `'off'`. Il valore predefinito è `'on'`.

```
node[:nginx][:gzip_static]
```

gzip_types

Elenco dei tipi MIME da comprimere (elenco di stringhe). Il valore predefinito è `['text/plain', 'text/html', 'text/css', 'application/x-javascript', 'text/xml', 'application/xml', 'application/xml+rss', 'text/javascript']`.

```
node[:nginx][:gzip_types]
```

gzip_vary

Indica se abilitare un'intestazione di risposta `Vary:Accept-Encoding` (stringa). I valori possibili sono `'on'` e `'off'`. Il valore predefinito è `'on'`.

```
node[:nginx][:gzip_vary]
```

keepalive

Indica se abilitare una connessione keep-alive (stringa). I valori possibili sono `'on'` e `'off'`. Il valore predefinito è `'on'`.

```
node[:nginx][:keepalive]
```

keepalive_timeout

Intervallo di tempo massimo, in secondi, durante il quale una connessione keep-alive rimane aperta (numero). Il valore predefinito è 65.

```
node[:nginx][:keepalive_timeout]
```

log_dir

Posizione dei file di log (stringa). Il valore predefinito è `'/var/log/nginx'`.

```
node[:nginx][:log_dir]
```

Utente

Utente (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `'www-data'`
- Ubuntu: `'nginx'`

```
node[:nginx][:user]
```

server_names_hash_bucket_size

Dimensione del bucket per le tabelle hash dei nomi di server, che può essere impostata su `32`, `64` o `128` (numero). Il valore predefinito è `64`.

```
node[:nginx][:server_names_hash_bucket_size]
```

worker_processes

Numero di processi di lavoro (numero). Il valore predefinito è `10`.

```
node[:nginx][:worker_processes]
```

worker_connections

Numero massimo di connessioni di lavoro (stringa). Il valore predefinito è `1024`. Il numero massimo di client è impostato su `worker_processes * worker_connections`.

```
node[:nginx][:worker_connections]
```

Attributi opsworks_berkshelf

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi attributi sono disponibili solo per gli stack Linux.

Gli attributi `opsworks_berkshelf` specificano la configurazione di Berkshelf. Per ulteriori informazioni, consulta Berkshelf. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

debug

Indica se includere le informazioni di debug di Berkshelf nel log di Chef (booleano). Il valore predefinito è `false`.

```
node['opsworks_berkshelf]['debug']
```

Attributi opsworks_java

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi attributi sono disponibili solo per gli stack Linux.

Gli attributi `opsworks_java` specificano la configurazione del server Tomcat. Per ulteriori informazioni, consulta la documentazione di riferimento relativa alla configurazione di Apache

Tomcat. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

datasources	java_app_server_version	java_shared_lib_dir
Attributi jvm_pkg	custom_pkg_location_url_debian	java_home_basedir
custom_pkg_location_url_rhel	use_custom_pkg_location	jvm_options
jvm_version	Attributi tomcat	

datasources

Set di attributi che definiscono i nomi delle risorse JNDI (stringa). Per ulteriori informazioni su come usare questo attributo, consulta Distribuzione di un'app JSP con un database di back-end. Il valore predefinito è un hash vuoto, che può essere sostituito con mappature personalizzate tra nomi brevi di app e nomi JNDI. Per ulteriori informazioni, consulta Distribuzione di un'app JSP con un database di back-end.

```
node['opsworks_java']['datasources']
```

java_app_server_version

Numero di versione del server di applicazioni Java (numero). Il valore predefinito è 7. Puoi sostituire questo attributo e specificare la versione 6. Se si installa un JDK non predefinito, questo attributo viene ignorato.

```
node['opsworks_java']['java_app_server_version']
```

java_shared_lib_dir

Directory delle librerie Java condivise (stringa). Il valore predefinito è `/usr/share/java`.

```
node['opsworks_java']['java_shared_lib_dir']
```

Attributi jvm_pkg

Set di attributi che puoi sostituire per installare un JDK non predefinito.

use_custom_pkg_location

Indica se installare un JDK personalizzato invece di OpenJDK (booleano). Il valore predefinito è `false`.

```
node['opsworks_java']['jvm_pkg']['use_custom_pkg_location']
```

custom_pkg_location_url_debian

Posizione del pacchetto JDK da installare sulle istanze Ubuntu (stringa). Il valore predefinito è `'http://aws.amazon.com/'`, che rappresenta un semplice valore di inizializzazione senza alcun significato specifico. Se desideri installare un JDK non predefinito, devi sostituire questo attributo e impostarlo sull'URL appropriato.

```
node['opsworks_java']['jvm_pkg']['custom_pkg_location_url_debian']
```

custom_pkg_location_url_rhel

Posizione del pacchetto JDK da installare sulle istanze Amazon Linux e RHEL (stringa). Il valore predefinito è `'http://aws.amazon.com/'`, che rappresenta un semplice valore di inizializzazione senza alcun significato specifico. Se desideri installare un JDK non predefinito, devi sostituire questo attributo e impostarlo sull'URL appropriato.

```
node['opsworks_java']['jvm_pkg']['custom_pkg_location_url_rhel']
```

java_home_basedir

Directory in cui verrà estratto il pacchetto JDK (stringa). Il valore predefinito è `/usr/local`. Non è necessario specificare questa impostazione per i pacchetti RPM, che includono già una struttura di directory completa.

```
node['opsworks_java']['jvm_pkg']['java_home_basedir']
```

jvm_options

Opzioni a riga di comando JVM, che consentono di specificare impostazioni quali la dimensione heap (stringa). Il set comune di opzioni è `-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`. Il valore predefinito non prevede alcuna opzione.

```
node['opsworks_java']['jvm_options']
```

jvm_version

Versione di OpenJDK (numero). Il valore predefinito è 7. Puoi sostituire questo attributo e specificare OpenJDK versione 6. Se si installa un JDK non predefinito, questo attributo viene ignorato.

```
node['opsworks_java']['jvm_version']
```

Attributi tomcat

Set di attributi che puoi sostituire per installare la configurazione Tomcat predefinita.

ajp_port	apache_tomcat_bind_mod	apache_tomcat_bind_path
auto_deploy	connection_timeout	mysql_connector_jar
port	secure_port	shutdown_port
threadpool_max_threads	threadpool_min_spare_threads	unpack_wars
uri_encoding	use_ssl_connector	use_threadpool
userdatabase_pathname		

ajp_port

Porta AJP (numero). Il valore predefinito è `8009`.

```
node['opsworks_java']['tomcat]['ajp_port']
```

apache_tomcat_bind_mod

Modulo proxy (stringa). Il valore predefinito è `proxy_http`. Puoi sostituire questo attributo per specificare il modulo proxy AJP `proxy_ajp`.

```
node['opsworks_java']['tomcat]['apache_tomcat_bind_mod']
```

apache_tomcat_bind_path

Percorso di associazione Apache-Tomcat (stringa). Il valore predefinito è `/`. Non è consigliabile sostituire questo attributo. La modifica del percorso di associazione potrebbe comportare il mancato funzionamento dell'applicazione.

```
node['opsworks_java']['tomcat]['apache_tomcat_bind_path']
```

auto_deploy

Indica se eseguire la distribuzione automatica (booleano). Il valore predefinito è `true`.

```
node['opsworks_java']['tomcat]['auto_deploy']
```

connection_timeout

Timeout della connessione, in millisecondi (numero). Il valore predefinito è `20000` (20 secondi).

```
node['opsworks_java']['tomcat]['connection_timeout']
```

mysql_connector_jar

File JAR della libreria di connettori MySQL (stringa). Il valore predefinito è `mysql-connector-java.jar`.

```
node['opsworks_java']['tomcat]['mysql_connector_jar']
```

port

Porta standard (numero). Il valore predefinito è `8080`.

```
node['opsworks_java']['tomcat]['port']
```

secure_port

Porta protetta (numero). Il valore predefinito è `8443`.

```
node['opsworks_java']['tomcat]['secure_port']
```

shutdown_port

Porta di chiusura (numero). Il valore predefinito è `8005`.

```
node['opsworks_java']['tomcat]['shutdown_port']
```

threadpool_max_threads

Numero massimo di thread nel pool di thread (numero). Il valore predefinito è `150`.

```
node['opsworks_java']['tomcat]['threadpool_max_threads']
```

threadpool_min_spare_threads

Numero minimo di thread liberi nel pool di thread (numero). Il valore predefinito è `4`.

```
node['opsworks_java']['tomcat]['threadpool_min_spare_threads']
```

unpack_wars

Indica se decomprimere i file WAR (booleano). Il valore predefinito è `true`.

```
node['opsworks_java']['tomcat]['unpack_wars']
```

uri_encoding

Codifica dell'URI (stringa). Il valore predefinito è `UTF-8`.

```
node['opsworks_java']['tomcat]['uri_encoding']
```

use_ssl_connector

Indica se utilizzare un connettore SSL (booleano). Il valore predefinito è `false`.

```
node['opsworks_java']['tomcat]['use_ssl_connector']
```

use_threadpool

Indica se utilizzare un pool di thread (booleano). Il valore predefinito è `false`.

```
node['opsworks_java']['tomcat]['use_threadpool']
```

userdatabase_pathname

Nome del percorso del database utente (stringa). Il valore predefinito è `conf/tomcat-users.xml`.

```
node['opsworks_java']['tomcat]['userdatabase_pathname']
```

Attributi passenger_apache2

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi attributi sono disponibili solo per gli stack Linux.

Gli attributi `passenger_apache2` specificano la configurazione di Phusion Passenger. Per ulteriori informazioni, consulta la Guida per l'utente di Phusion Passenger, versione Apache. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

friendly_error_pages	gem_bin	gems_path
high_performance_mode	root_path	max_instances_per_app
max_pool_size	max_requests	module_path
pool_idle_time	rails_app_spawner_idle_time	rails_framework_spawner_idle_time
rails_spawn_method	ruby_bin	ruby_wrapper_bin

stat_throttle_rate	version	

friendly_error_pages

Indica se visualizzare una pagina di errore intuitiva in caso di mancato avvio dell'applicazione (stringa). Questo attributo può essere impostato su "on" e "off"; il valore predefinito è "off".

```
node[:passenger][:friendly_error_pages]
```

gem_bin

Posizione dei file binari del pacchetto gem (stringa). Il valore predefinito è `'/usr/local/bin/gem'`.

```
node[:passenger][:gem_bin]
```

gems_path

Percorso dei pacchetti gem (stringa). Il valore predefinito dipende dalla versione di Ruby. Per esempio:

- Ruby versione 1.8: `'/usr/local/lib/ruby/gems/1.8/gems'`
- Ruby versione 1.9: `'/usr/local/lib/ruby/gems/1.9.1/gems'`

```
node[:passenger][:gems_path]
```

high_performance_mode

Indica se utilizzare la modalità a prestazioni elevate di Passenger (stringa). I valori possibili sono `'on'` e `'off'`. Il valore predefinito è `'off'`.

```
node[:passenger][:high_performance_mode ]
```

root_path

Directory radice di Passenger (stringa). Il valore predefinito dipende dalle versioni di Ruby e Passenger. Nella sintassi di Chef, il valore è `"#{node[:passenger][:gems_path]}/passenger-#{passenger[:version]}"`.

```
node[:passenger][:root_path]
```

max_instances_per_app

Numero massimo di processi applicazione per applicazione (numero). Il valore predefinito è `0`. Per ulteriori informazioni, consulta PassengerMaxInstancesPerApp.

```
node[:passenger][:max_instances_per_app]
```

max_pool_size

Numero massimo di processori dell'applicazione (numero). Il valore predefinito è `8`. Per ulteriori informazioni, consulta PassengerMaxPoolSize.

```
node[:passenger][:max_pool_size]
```

max_requests

Numero massimo di richieste (numero). Il valore predefinito è `0`.

```
node[:passenger][:max_requests]
```

module_path

Percorso del modulo (stringa). I valori predefiniti sono riportati di seguito:

- Amazon Linux e RHEL: `"#{node['apache']['libexecdir']}/mod_passenger.so"`
- Ubuntu: `"#{passenger[:root_path]}/ext/apache2/mod_passenger.so"`

```
node[:passenger][:module_path]
```

pool_idle_time

Tempo massimo, in secondi, durante il quale un processo dell'applicazione può rimanere inattivo (numero). Il valore predefinito è `14400` (4 ore). Per ulteriori informazioni, consulta PassengerPoolIdleTime.

```
node[:passenger][:pool_idle_time]
```

rails_app_spawner_idle_time

Intervallo massimo di inattività per il generatore dell'app Rails (numero). Se questo attributo è impostato su zero, non si verifica mai il timeout del generatore dell'app. Il valore predefinito è `0`. Per ulteriori informazioni, consulta l'argomento relativo ai metodi di generazione.

```
node[:passenger][:rails_app_spawner_idle_time]
```

rails_framework_spawner_idle_time

Intervallo massimo di inattività per il generatore del framework Rails (numero). Se questo attributo è impostato su zero, non si verifica mai il timeout del framework. Il valore predefinito è `0`. Per ulteriori informazioni, consulta l'argomento relativo ai metodi di generazione.

```
node[:passenger][:rails_framework_spawner_idle_time]
```

rails_spawn_method

Metodo di generazione Rails (stringa). Il valore predefinito è `'smart-lv2'`. Per ulteriori informazioni, consulta l'argomento relativo ai metodi di generazione.

```
node[:passenger][:rails_spawn_method]
```

ruby_bin

Posizione dei file binari Ruby (stringa). Il valore predefinito è `'/usr/local/bin/ruby'`.

```
node[:passenger][:ruby_bin]
```

ruby_wrapper_bin

Posizione degli script wrapper Ruby (stringa). Il valore predefinito è `'/usr/local/bin/ruby_gc_wrapper.sh'`.

```
node[:passenger][:ruby_wrapper_bin]
```

stat_throttle_rate

Velocità in base alla quale Passenger esegue i controlli del file system (numero). Il valore predefinito è 5, ovvero i controlli vengono eseguiti al massimo una volta ogni 5 secondi. Per ulteriori informazioni, consulta PassengerStatThrottleRate .

```
node[:passenger][:stat_throttle_rate]
```

version

Versione (stringa). Il valore predefinito è `'3.0.9'`.

```
node[:passenger][:version]
```

Attributi ruby

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi attributi sono disponibili solo per gli stack Linux.

Gli attributi `ruby` specificano la versione di Ruby utilizzata dalle applicazioni. L'utilizzo degli attributi è stato modificato con l'introduzione della funzione Versioni multiple della semantica in Ruby 2.1. Per ulteriori informazioni su come specificare una versione, inclusi gli esempi, consulta Versioni di Ruby. Per i dettagli completi su come AWS OpsWorks Stacks determina la versione di Ruby, consulta il file degli attributi integrato, ruby.rb. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

full_version

Numero di versione completo (stringa). Non è consigliabile sostituire questo attributo. Utilizza invece [:opsworks][:ruby_version] e l'attributo appropriato relativo alla versione della patch per specificare una versione.

```
[:ruby][:full_version]
```

major_version

Numero della versione principale (stringa). Non è consigliabile sostituire questo attributo. Utilizza invece [:opsworks][:ruby_version] per specificare la versione principale.

```
[:ruby][:major_version]
```

minor_version

Numero della versione secondaria (stringa). Non è consigliabile sostituire questo attributo. Utilizza invece [:opsworks][:ruby_version] per specificare la versione secondaria.

```
[:ruby][:minor_version]
```

patch

Livello della patch (string). Questo attributo è valido per Ruby versione 2.0.0 e precedenti. Per le versioni successive di Ruby, utilizza l'attributo `patch_version`.

```
[:ruby][:patch]
```

Il numero di patch deve essere preceduto da p. Ad esempio, puoi utilizzare il JSON personalizzato per specificare il livello di patch 484.

```
{
  "ruby":{"patch":"p484"}
}
```

patch_version

Numero della patch (string). Questo attributo è valido per Ruby versione 2.1 e successive Per le versioni precedenti di Ruby, utilizza l'attributo `patch`.

```
[:ruby][:patch_version]
```

pkgrelease

Numero di versione del pacchetto (stringa).

```
[:ruby][:pkgrelease]
```

Attributi unicorn

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questi attributi sono disponibili solo per gli stack Linux.

Gli attributi `unicorn` specificano la configurazione di Unicorn. Per ulteriori informazioni, consulta Unicorn::Configurator. Per ulteriori informazioni su come sostituire gli attributi predefiniti per specificare valori personalizzati, consulta Sostituzione degli attributi.

accept_filter	backlog	delay
tcp_nodelay	tcp_nopush	preload_app
timeout	tries	version
worker_processes		

accept_filter

Filtro di accettazione, `'httpready'` o `'dataready'` (stringa). Il valore predefinito è `'httpready'`.

```
node[:unicorn][:accept_filter]
```

backlog

Numero massimo di richieste che la coda può contenere (numero). Il valore predefinito è `1024`.

```
node[:unicorn][:backlog]
```

delay

Intervallo di tempo, in secondi, di attesa dei nuovi tentativi di associazione di un socket (numero). Il valore predefinito è `0.5`.

```
node[:unicorn][:delay]
```

preload_app

Indica se precaricare un'app prima di eseguire il fork di un processo di lavoro (booleano). Il valore predefinito è `true`.

```
node[:unicorn][:preload_app]
```

tcp_nodelay

Indica se disabilitare l'algoritmo di Nagle per i socket TCP (booleano). Il valore predefinito è `true`.

```
node[:unicorn][:tcp_nodelay]
```

tcp_nopush

Indica se abilitare TCP_CORK (booleano). Il valore predefinito è `false`.

```
node[:unicorn][:tcp_nopush]
```

timeout

Intervallo di tempo massimo, in secondi, che un processo di lavoro può utilizzare per ogni richiesta (numero). I processi di lavoro che superano il valore di timeout vengono terminati. Il valore predefinito è `60`.

```
node[:unicorn][:timeout]
```

tries

Numero massimo di nuovi tentativi di associazione a un socket (numero). Il valore predefinito è `5`.

```
node[:unicorn][:tries]
```

version

Versione di Unicorn (stringa). Il valore predefinito è `'4.7.0'`.

```
node[:unicorn][:version]
```

worker_processes

Numero di processi di lavoro (numero). Il valore predefinito è `max_pool_size`, se esistente. In caso contrario, è `4`.

```
node[:unicorn][:worker_processes]
```

Risoluzione dei problemi di Chef 11.10 e delle versioni precedenti per Linux

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Note

Per ulteriori informazioni relative alla risoluzione di problemi, consulta Guida al debugging e alla risoluzione dei problemi.

Log di Chef per Chef 11.10 e per le versioni precedenti per Linux

AWS OpsWorks Stacks memorizza i log di Chef di ogni istanza nella sua directory. `/var/lib/aws/opsworks/chef` È necessario disporre di privilegi sudo per accedere a questa directory. Il log per ogni esecuzione si trova in un file chiamato *YYYY*-*MM*-*DD*-*HH*-*MM*-*SS*-*NN*.log.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Visualizzazione di un log di Chef con la console
- Visualizzazione di un log di Chef tramite la CLI o l'API
- Interpretazione di un log di Chef
- Errori comuni del log di Chef

Utilizzo di AWS OpsWorks Stacks con altri servizi AWS

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Puoi fare in modo che i server delle applicazioni in esecuzione in uno AWS OpsWorks stack Stacks utilizzino una varietà di servizi AWS che non sono direttamente integrati con AWS OpsWorks Stacks. Ad esempio, puoi fare in modo che i tuoi server applicativi utilizzino Amazon RDS come database di back-end. È possibile accedere a tali servizi utilizzando il seguente schema generale:

1. Creare e configurare il servizio AWS utilizzando la console AWS, l'API o l'interfaccia a riga di comando e registrare i dati di configurazione che saranno necessari all'applicazione per accedere al servizio, ad esempio nome host o porta.
2. Creare una o più ricette personalizzate per configurare l'applicazione in modo che possa accedere al servizio.

   La ricetta ottiene i dati di configurazione dagli attributi JSON di configurazione e distribuzione dello stack definiti tramite codice JSON personalizzato prima dell'esecuzione delle ricette.
3. Assegnare la ricetta personalizzata all'evento del ciclo di vita Deploy (Distribuzione) nel livello del server applicazioni.
4. Creare un oggetto JSON personalizzato che assegna i valori appropriati agli attributi dei dati di configurazione e aggiungerlo al codice JSON di configurazione e distribuzione dello stack.
5. Distribuire l'applicazione nello stack.

Il processo di distribuzione esegue ricette personalizzate, che utilizzano i valori dei dati di configurazione definiti nel codice JSON personalizzato per configurare l'applicazione in modo che possa accedere al servizio.

Questa sezione descrive come fare in modo che i server applicativi AWS OpsWorks Stacks accedano a una varietà di servizi AWS. Presuppone che tu abbia già familiarità con i libri di ricette Chef e il modo in cui le ricette possono utilizzare gli attributi JSON di configurazione e dello stack per configurare le applicazioni, in genere creando file di configurazione. In caso contrario, consulta prima Libri di ricette e ricette e Personalizzazione degli stack AWS OpsWorks.

Argomenti

- Utilizzo di un archivio dati di back-end
- Utilizzo di ElastiCache Redis come archivio chiave-valore in memoria
- Utilizzo di un bucket Amazon S3
- Utilizzo AWS CodePipeline con AWS OpsWorks Stacks

Utilizzo di un archivio dati di back-end

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Gli stack di server applicativi includono in genere un server di database per fornire un data store di back-end. AWS OpsWorks Stacks fornisce supporto integrato per i server MySQL tramite il livello MySQL e per diversi tipi di server di database tramite il livello Amazon Relational Database Service (Amazon RDS). Tuttavia, puoi personalizzare facilmente uno stack in modo che i server delle applicazioni utilizzino altri server di database come Amazon DynamoDB o MongoDB. Questo argomento descrive la procedura di base per la connessione di un server di applicazione a un server di database AWS. Utilizza lo stack e l'applicazione da Nozioni di base sugli stack Linux Chef 11 per mostrare come connettere manualmente un server di applicazione PHP a un database RDS.

Anche se l'esempio è basato su uno stack di Linux, i principi di base, inoltre, si applicano agli stack Windows. Per un esempio di come incorporare un server di database MongoDB in uno stack, vedi Deploying MongoDB with. OpsWorks

> **Note**
>
> Questo argomento utilizza Amazon RDS come esempio pratico. Tuttavia, se desideri utilizzare un database Amazon RDS con il tuo stack, è molto più semplice utilizzare un livello Amazon RDS.

Argomenti

- Come impostare una connessione di database
- Come connettere un'istanza di Application Server ad Amazon RDS

Come impostare una connessione di database

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Configurare la connessione tra un server di applicazione e il suo database di back-end utilizzando una ricetta personalizzata. La ricetta configura il server di applicazione come richiesto, in genere mediante la creazione di un file di configurazione. La ricetta ottiene i dati di connessione, come il nome dell'host e del database, da un set di attributi negli attributi di configurazione e distribuzione dello stack che AWS OpsWorks Stacks installa su ogni istanza.

Ad esempio, Step 2 of si Nozioni di base sugli stack Linux Chef 11 basa su uno stack denominato MyStack con due livelli, PHP App Server e MySQL, ciascuno con un'istanza. Si distribuisce un'app denominata PHPApp Simple nell'istanza PHP App Server che utilizza il database sull'istanza MySQL come archivio dati di back-end. Quando si distribuisce l'applicazione, AWS OpsWorks Stacks installa la configurazione stack e gli attributi di distribuzione che contengono le informazioni di connessione

al database. L'esempio seguente mostra gli attributi di connessione al database, rappresentati come JSON:

```
{
  ...
  "deploy": {
    "simplephpapp": {
      ...
      "database": {
        "reconnect": true,
        "password": null,
        "username": "root",
        "host": null,
        "database": "simplephpapp"
        ...
      },
      ...
    }
  }
}
```

I valori degli attributi sono forniti da AWS OpsWorks Stacks e sono generati o basati sulle informazioni fornite dall'utente.

Per consentire a Simple di accedere PHPApp al data store, è necessario configurare la connessione tra il server delle applicazioni PHP e il database MySQL assegnando una ricetta personalizzata `appsetup.rb` denominata all'evento Deploy del ciclo di vita Deploy del livello PHP App Server. Quando si distribuisce SimplePHPApp, viene eseguito AWS OpsWorks Stacks`appsetup.rb`, che crea un file di configurazione denominato `db-connect.php` che configura la connessione, come illustrato nel seguente estratto.

```
node[:deploy].each do |app_name, deploy|
  ...
  template "#{deploy[:deploy_to]}/current/db-connect.php" do
    source "db-connect.php.erb"
    mode 0660
    group deploy[:group]

    if platform?("ubuntu")
```

```
      owner "www-data"
    elsif platform?("amazon")
      owner "apache"
    end

    variables(
      :host =>     (deploy[:database][:host] rescue nil),
      :user =>     (deploy[:database][:username] rescue nil),
      :password => (deploy[:database][:password] rescue nil),
      :db =>       (deploy[:database][:database] rescue nil),
      :table =>    (node[:phpapp][:dbtable] rescue nil)
    )
    ...
  end
end
```

Le variabili che caratterizzano la connessione, e così via `hostuser`, vengono impostate con i valori corrispondenti dagli attributi JSON di deploy. `[:deploy][:app_name][:database]` Per semplicità, l'esempio presuppone sia già stata creata una tabella denominata `urler`, quindi il nome della tabella è rappresentato da `[:phpapp][:dbtable]` nel file attributi del libro di cucina.

Questa ricetta può effettivamente connettere il server delle applicazioni PHP a qualsiasi server di database MySQL, non solo ai membri di un livello MySQL. Per utilizzare un server MySQL diverso, devi solo impostare gli attributi su valori appropriati per `[:database]` il tuo server, cosa che puoi fare utilizzando JSON personalizzato. AWS OpsWorks Stacks incorpora quindi tali attributi e valori negli attributi di configurazione e distribuzione dello stack e li `appsetup.rb` utilizza per creare il modello che configura la connessione. Per ulteriori informazioni sulla sostituzione della configurazione stack e implementazione JSON, consulta Sostituzione degli attributi.

Come connettere un'istanza di Application Server ad Amazon RDS

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione descrive come personalizzare il MyStack modulo per connettere il server delle applicazioni PHP a un'istanza RDS. Nozioni di base sugli stack Linux Chef 11

Argomenti

- Crea un database Amazon RDS MySQL
- Personalizzare lo Stack per la connessione al database RDS

Crea un database Amazon RDS MySQL

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Ora sei pronto per creare un database RDS per l'esempio utilizzando il Launch DB Instance Wizard della console Amazon RDS. La procedura seguente è un breve riepilogo dei principali dettagli. Per una descrizione dettagliata di come creare un database, consulta Nozioni di base su Amazon RDS.

Per creare il database Amazon RDS

1. Se è la prima volta che si crea un database RDS, fare clic su Get Started Now (Inizia subito). In caso contrario, fare clic su RDS Dashboard (Pannello di controllo RDS) nel riquadro di navigazione e fare clic su Launch a DB Instance (Avvia un'istanza di database).
2. Selezionare MySQL Community Edition (Edizione community MySQL) come istanza database.
3. Per Do you plan to use this database for production purposes? (Intendi utilizzare questo database per scopi di produzione?) selezionare No, this instance... (No, questa istanza...), che è sufficiente per l'esempio. Per l'uso nella produzione, è consigliabile selezionare Yes, use Multi-AZ Deployment... (Sì, utilizzare implementazione Multi-AZ...). Fare clic su Next Step (Fase successiva).
4. Nella pagina Specify DB Details (Specifica dettagli database), specificare le seguenti impostazioni:

    - DB Instance Class (Classe di istanza database): db.t2.micro

- Multi-AZ deployment (Implementazione Multi-AZ): No
- Allocated Storage (Storage allocato): **5** GB
- DB Instance Identifier (Identificatore istanze DB): **rdsexample**
- Master Username (Nome utente master): **opsworksuser**.
- Master Password (Password master): specificare una password adatta e registrarla per l'utilizzo in un secondo momento.

Accettare le impostazioni predefinite per le altre opzioni e fare clic su Next Step (Fase successiva).

5. Nella pagina Configure Advanced Settings (Configura impostazioni avanzate), specificare le seguenti impostazioni:

   - Nella sezione Network & Security (Rete e sicurezza), per VPC Security Group(s) (Gruppi di sicurezza VPC), selezionare phpsecgroup (VPC)
   - Nella sezione Database Options (Opzioni database), per Database Name (Nome del database), digitare **rdsexampledb**.
   - Nella sezione Backup, impostare Backup Retention Period (Periodo di ritenzione dei backup) su 0 ai fini di questo scenario.

   Accettare le impostazioni predefinite per le altre opzioni e fare clic su Launch DB Instance (Avvia istanza database).
6. Scegliere View Your DB Instances (Visualizza istanze database) per visualizzare un elenco di istanze database.
7. Selezionare l'istanza rdsexample nell'elenco e fare clic sulla freccia per rivelare l'endpoint dell'istanza e altri dettagli. Registrare l'endpoint da utilizzare in futuro. Avrà un nome tipo `rdsexample.c6c8mntzhgv0.us-west-2.rds.amazonaws.com:3306`. Basta registrare il nome DNS; non è necessario il numero di porta.
8. Utilizzare uno strumento come MySQL Workbench per creare una tabella denominata `urler` nel database `rdsexampledb` utilizzando il seguente comando SQL:

```
CREATE TABLE urler(id INT UNSIGNED NOT NULL AUTO_INCREMENT,author VARCHAR(63) NOT
 NULL,message TEXT,PRIMARY KEY (id))
```

Personalizzare lo Stack per la connessione al database RDS

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Dopo aver creato un'istanza RDS da utilizzare come database di back-end per il server delle applicazioni PHP, puoi personalizzare da. MyStack Nozioni di base sugli stack Linux Chef 11

Per connettere il server PHP App a un database RDS

1. Apri la console AWS OpsWorks Stacks e crea uno stack con un livello PHP App Server che contenga un'istanza e distribuisci Simple, come descritto in. PHPApp Nozioni di base sugli stack Linux Chef 11 Questo stack utilizza la versione 1 di SimplePHPApp, che non utilizza una connessione al database.
2. Aggiornare la configurazione dello stack per utilizzare i libri di cucina personalizzati che includono la ricetta `appsetup.rb` e i relativi file di modello e attributo.

    1. Impostare Use custom Chef Cookbooks (Utilizza i libri di ricette di Chef personalizzati) su Yes (Sì).
    2. Impostare Repository type (Tipo archivio) su Git e Repository URL (Archivio URL) su `git://github.com/amazonwebservices/opsworks-example-cookbooks.git`.
3. Aggiungere alla casella dello stack Custom Chef JSON (Chef JSON personalizzato) per assegnare i dati di connessione RDS agli attributi `[:database]` che `appsetup.rb` utilizza per creare il file di configurazione.

```
{
  "deploy": {
    "simplephpapp": {
      "database": {
        "username": "opsworksuser",
        "password": "your_password",
        "database": "rdsexampledb",
```

```
        "host": "rds_endpoint",
        "adapter": "mysql"
      }
    }
  }
}
```

Utilizzare i seguenti valori di attributo:

- username (nome utente): il nome utente master specificato durante la creazione dell'istanza RDS.

  Questo esempio usa `opsworksuser`.
- password: la password master specificata durante la creazione dell'istanza RDS.

  Inserire la password specificata.
- database: il database creato durante la creazione dell'istanza RDS.

  Questo esempio usa `rdsexampledb`.
- host: l'endpoint dell'istanza RDS, ottenuta dalla console di RDS durante la creazione dell'istanza nella sezione precedente. Non includere il numero di porta.
- adapter (adattatore): l'adattatore.

  L'istanza RDS per questo esempio utilizza MySQL, perciò adapter (adattatore) è impostato su `mysql`. A differenza degli altri attributi, adapter (adattatore) non viene utilizzato da `appsetup.rb`. Viene invece utilizzato dalla ricetta Configure integrata nel livello PHP App Server per creare un file di configurazione diverso.

4. Modifica la PHPApp configurazione Simple per specificare una versione di Simple PHPApp che utilizza un database di back-end, come segue:

   - Document root (Documento root): impostare questa opzione su `web`.
   - Branch/Revision (Ramo/Revisione): impostare questa opzione su `version2`.

   Lasciare le opzioni rimanenti invariate.
5. Modifica il livello PHP App Server per configurare la connessione al database aggiungendole `phpapp::appsetup` alle ricette Deploy del livello.
6. Implementa la nuova versione Simple. PHPApp

7. Quando Simple PHPApp è distribuito, esegui l'applicazione andando alla pagina Istanze e facendo clic sull'indirizzo IP pubblico dell'istanza php-app1. Dovrebbe apparire la pagina seguente nel browser, che consente di immettere testo e memorizzarlo nel database.



> **Note**
>
> Se lo stack ha un livello MySQL AWS OpsWorks , Stacks assegna automaticamente i dati di connessione corrispondenti agli attributi. `[:database]` Tuttavia, se si assegna JSON personalizzato allo stack che definisce valori diversi `[:database]`, sostituiscono i valori predefiniti. Poiché gli `[:deploy]` attributi sono installati su ogni istanza, tutte le ricette che dipendono dagli `[:database]` attributi utilizzeranno i dati di connessione personalizzati, non i dati del livello MySQL per. Se si desidera che un determinato livello di server di applicazione utilizzi dati di connessione personalizzati, assegnare JSON personalizzato all'evento di distribuzione del livello e limitare tale distribuzione a quel livello. Per ulteriori informazioni su come utilizzare gli attributi di distribuzione, consulta Distribuzione di app. Per ulteriori informazioni sulla sostituzione di attributi integrati AWS OpsWorks Stacks, consulta Sostituzione degli attributi.

Utilizzo di ElastiCache Redis come archivio chiave-valore in memoria

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> Questo argomento è basato su uno stack Linux, ma gli stack di Windows possono anche utilizzare Amazon ElastiCache ()ElastiCache. Per un esempio di utilizzo ElastiCache con un'istanza di Windows, vedi ElastiCache ASP.NET Session Store.

Spesso è possibile migliorare le prestazioni del server delle applicazioni utilizzando un server di memorizzazione nella cache per fornire un archivio chiave-valore in memoria per piccoli elementi di dati come le stringhe. Amazon ElastiCache è un servizio AWS che semplifica la fornitura di supporto per la memorizzazione nella cache per il server delle applicazioni, utilizzando i motori di caching Memcached o Redis. AWS OpsWorks Stacks fornisce supporto integrato per Memcached. Tuttavia, se Redis soddisfa meglio le vostre esigenze, potete personalizzare lo stack in modo che i server delle applicazioni utilizzino Redis. ElastiCache

Questo argomento illustra il processo di base per fornire il supporto per la memorizzazione nella cache ElastiCache Redis per gli stack Linux, utilizzando un server di applicazioni Rails come esempio. Presuppone che tu disponga già di un'apposita applicazione Ruby on Rails. Per ulteriori informazioni su ElastiCache, consulta What Is Amazon ElastiCache? .

Argomenti

- Fase 1: Creare un cluster ElastiCache Redis
- Fase 2: impostazione di uno stack Rails
- Fase 3: creazione e distribuzione di un libro di ricette personalizzato
- Fase 4: Assegna la ricetta a un evento LifeCycle

- Fase 5: aggiungere informazioni di accesso al JSON di configurazione dello stack
- Fase 6: distribuzione ed esecuzione dell'app

Fase 1: Creare un cluster ElastiCache Redis

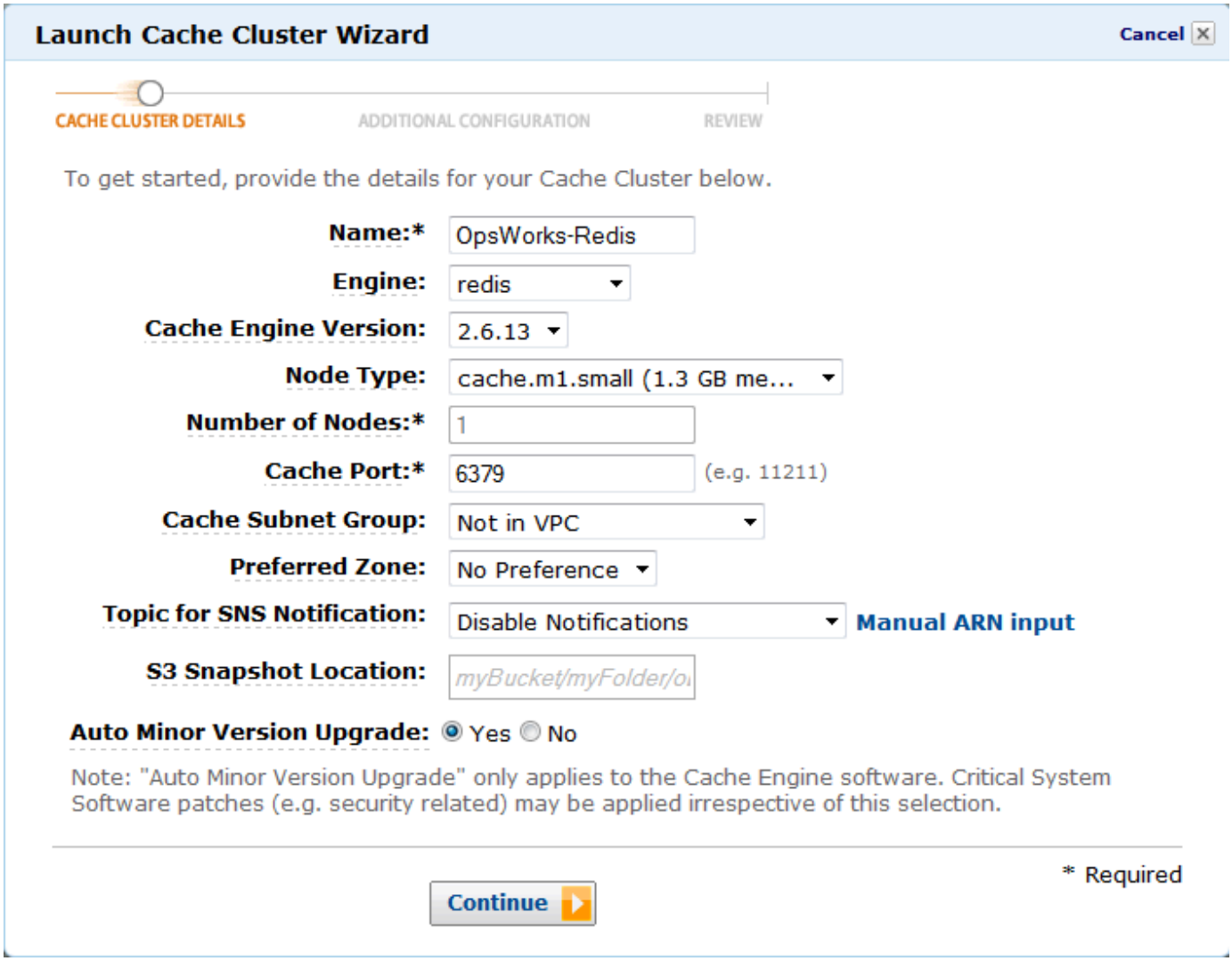
> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Devi prima creare un cluster Amazon ElastiCache Redis utilizzando la ElastiCache console, l'API o la CLI. Di seguito è illustrato come utilizzare la console per creare un cluster.

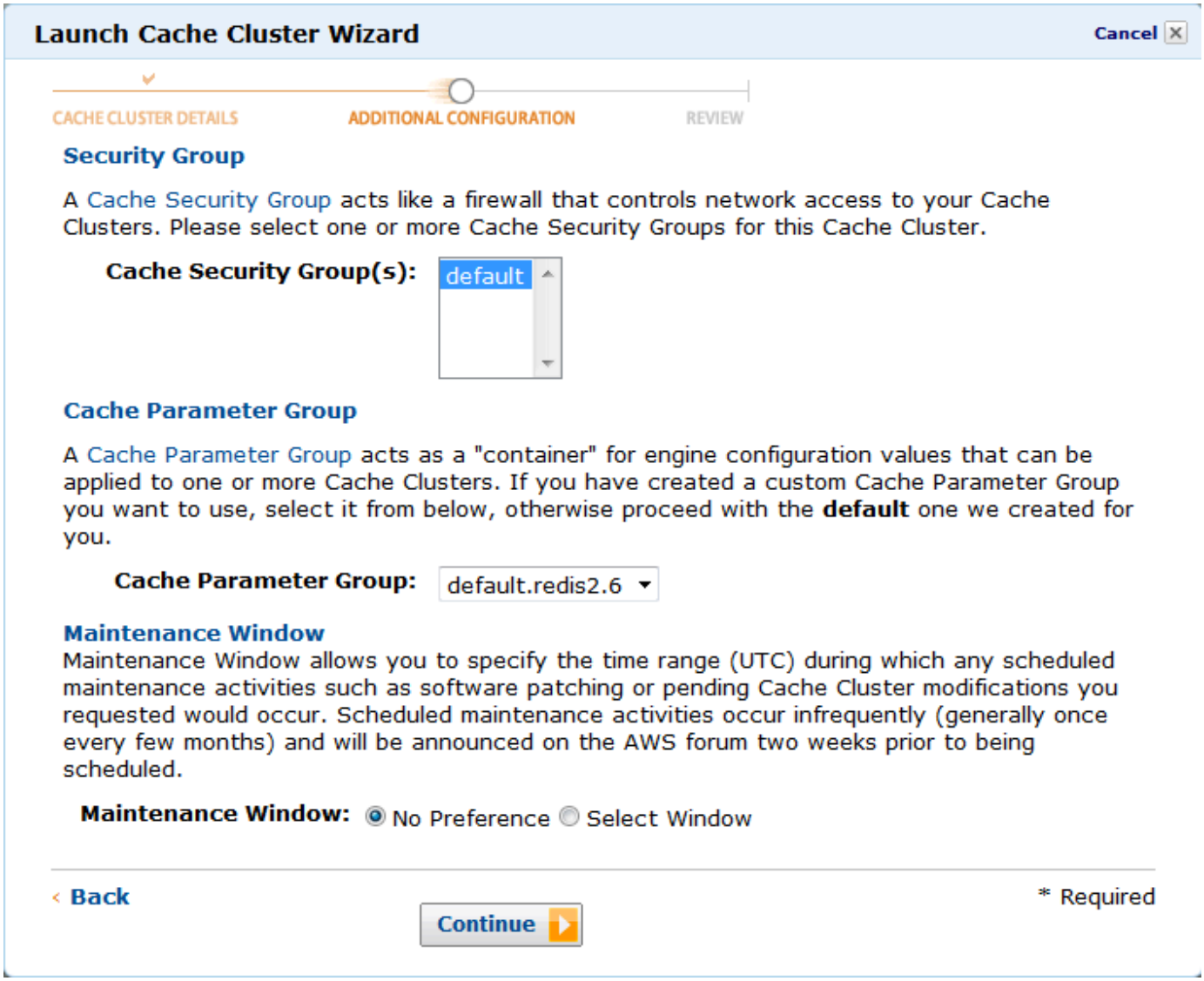
Per creare un cluster Redis ElastiCache

1. Aprire la console ElastiCache e fare clic su Launch Cache Cluster (Avvia Cluster di cache) per avviare la procedura guidata di Cluster di cache.
2. Nella pagina dei dettagli del Cluster di cache, seguire questi passaggi:
   - Impostare l'opzione Name (Nome) sul nome del server di cache.

     Questo esempio utilizza OpsWorks -Redis.
   - Impostare l'opzione Engine (Motore) su redis.
   - Impostare Topic for SNS Notification (Argomento per notifica SNS) su Disable Notifications (Disabilita notifiche).
   - Accettare i valori predefiniti per le altre impostazioni e fare clic su Continue (Continua).



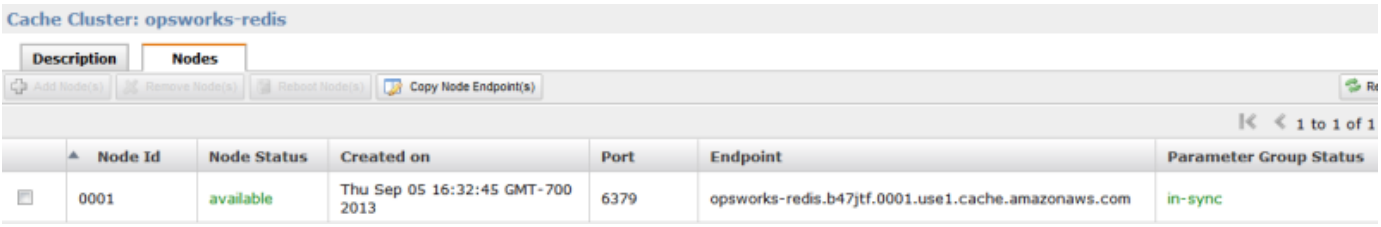
3. Nella pagina Additional Configuration (Configurazione aggiuntiva), accettare le impostazioni predefinite e fare clic su Continue (Continua).



4. Fare clic su Launch Cache Cluster (Avvia Cluster di cache) per creare il cluster.

> **Important**
>
> Il gruppo di sicurezza cache di default è sufficiente per questo esempio, ma per l'uso negli ambienti di produzione è necessario crearne uno appropriato. Per ulteriori informazioni, consulta la pagina sulla gestione dei gruppi di sicurezza cache.

5. Dopo l'avvio del cluster, fare clic sul nome per aprire la pagina dei dettagli, quindi fare clic sulla scheda Nodes (Nodi). Registrare l'opzione Port (Porta) del cluster e i valori Endpoint per un uso successivo.



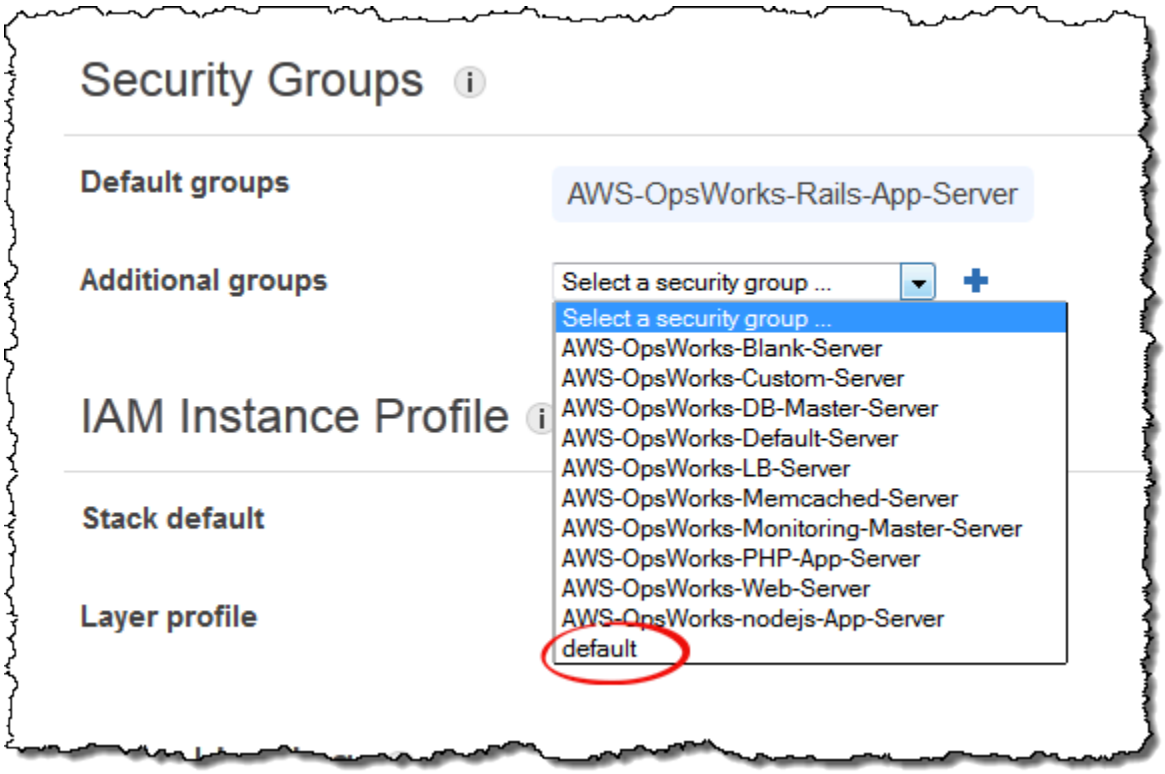
Fase 2: impostazione di uno stack Rails

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Oltre a creare uno stack che supporti un livello Rails App Server, devi anche configurare i gruppi di sicurezza del livello in modo che il server Rails possa comunicare correttamente con il server Redis.

Per configurare uno stack

1. Crea un nuovo stack, denominato in questo **`RedisStack`** esempio, e aggiungi un livello Rails App Server. È possibile utilizzare le impostazioni di default per entrambi. Per ulteriori informazioni, consulta Creare un nuovo stack e Creazione di un livello OpsWorks .
2. Nella pagina Livelli, per Rails App Server, fai clic su Sicurezza, quindi su Modifica.
3. Accedere alla sezione Security Groups (Gruppi di sicurezza) e aggiungere il gruppo di sicurezza del cluster ElastiCache all'opzione Additional groups (Altri gruppi). Per questo esempio, selezionare il gruppo di sicurezza di default, fare clic su + per aggiungerlo al livello, quindi fare clic su Save (Salva) per salvare la nuova configurazione.



4. Aggiungi un'istanza al livello Rails App Server e avviala. Per ulteriori informazioni su come aggiungere e avviare istanze, consulta Aggiunta di un'istanza a un livello.

Fase 3: creazione e distribuzione di un libro di ricette personalizzato

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Allo stato attuale, lo stack non è ancora pienamente funzionale; devi abilitare l'applicazione per accedere al server Redis. L'approccio più flessibile è inserire un file YAML con le informazioni di accesso nella sottocartella `config` dell'applicazione. L'applicazione è quindi in grado di recuperare le informazioni dal file. Con questo approccio, puoi modificare le informazioni di connessione senza riscrivere e ridistribuire l'applicazione. Per questo esempio, il file deve essere denominato

`redis.yml` e deve contenere il nome host e la porta del cluster ElastiCache , come indicato di seguito:

```
host: cache-cluster-hostname
port: cache-cluster-port
```

Puoi copiare manualmente questo file sui tuoi server, ma un approccio migliore consiste nell'implementare una ricetta Chef per generare il file e fare in modo che AWS OpsWorks Stacks esegua la ricetta su ogni server. Le ricette Chef sono applicazioni Ruby specializzate che AWS OpsWorks Stacks utilizza per eseguire attività su istanze come l'installazione di pacchetti o la creazione di file di configurazione. Le ricette sono contenute in un libro di ricette, che può contenere più ricette e i file correlati, ad esempio i modelli per i file di configurazione. Il ricettario viene inserito in un repository, ad esempio GitHub, e deve avere una struttura di directory standard. Se non disponi già di un repository personalizzato per il libro di ricette, consulta Repository dei libri di ricette per informazioni su come crearne uno.

Per questo esempio, aggiungi un libro di ricette denominato `redis-config` al relativo repository con il seguente contenuto:

```
my_cookbook_repository
  redis-config
    recipes
      generate.rb
    templates
      default
        redis.yml.erb
```

La cartella `recipes` contiene una ricetta denominata `generate.rb`, che genera il file di configurazione dell'applicazione da `redis.yml.erb`, come indicato di seguito:

```
node[:deploy].each do |app_name, deploy_config|
  # determine root folder of new app deployment
  app_root = "#{deploy_config[:deploy_to]}/current"

  # use template 'redis.yml.erb' to generate 'config/redis.yml'
  template "#{app_root}/config/redis.yml" do
    source "redis.yml.erb"
```

```
    cookbook "redis-config"

    # set mode, group and owner of generated file
    mode "0660"
    group deploy_config[:group]
    owner deploy_config[:user]

    # define variable “@redis” to be used in the ERB template
    variables(
      :redis => deploy_config[:redis] || {}
    )

    # only generate a file if there is Redis configuration
    not_if do
      deploy_config[:redis].blank?
    end
  end
end
```

La ricetta dipende dai dati dell'oggetto JSON di configurazione e distribuzione dello stack AWS OpsWorks Stacks, che viene installato su ogni istanza e contiene informazioni dettagliate sullo stack e sulle app distribuite. Il nodo `deploy` dell'oggetto ha la seguente struttura:

```
{
   ...
  "deploy": {
    "app1": {
      "application" : "short_name",
      ...
    }
    "app2": {
      ...
    }
    ...
  }
}
```

Il nodo di distribuzione contiene un set di oggetti JSON incorporati, uno per ogni applicazione distribuita, denominati con il nome breve dell'app. Ogni oggetto dell'app contiene un set di attributi che definiscono la configurazione dell'app, ad esempio la radice del documento e il

tipo di applicazione. Per un elenco degli attributi di distribuzione, consulta Attributi deploy. Le ricette possono utilizzare la sintassi degli attributi di Chef per rappresentare i valori del JSON di configurazione e distribuzione dello stack. Ad esempio, `[:deploy][:app1][:application]` rappresenta il nome breve dell'applicazione app1.

Per ogni applicazione `[:deploy]`, la ricetta esegue il blocco di codice associato, dove `deploy_config` rappresenta l'attributo dell'app. La ricetta prima imposta `app_root` nella directory principale dell'app, `[:deploy][:app_name][:deploy_to]/current`, Quindi, utilizza un modello di risorsa Chef per generare un file di configurazione da `redis.yml.erb` e inserirlo in `app_root/config`.

I file di configurazione sono in genere creati da modelli, con molte impostazioni definite dagli attributi di Chef. Con gli attributi puoi modificare le impostazioni tramite JSON personalizzati, come descritto più avanti, invece di riscrivere il file modello. Il modello `redis.yml.erb` contiene quanto segue:

```
host: <%= @redis[:host] %>
port: <%= @redis[:port] || 6379 %>
```

Gli elementi <%... %> sono segnaposti che rappresentano un valore attributo.

- `<%= @redis[:host] %>` rappresenta il valore di `redis[:host]`, che è il nome host del cluster di cache.
- `<%= @redis[:port] || 6379 %>` rappresenta il valore di `redis[:port]` oppure, se tale attributo non è definito, il valore della porta di default, 6379.

La risorsa `template` funziona come segue:

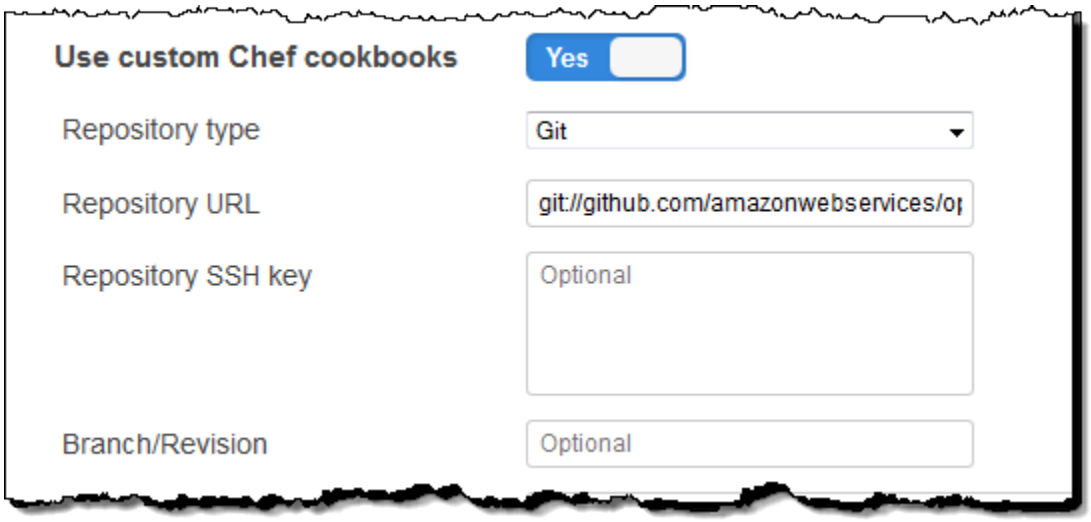
- `source` e `cookbook` specificano rispettivamente il modello e i nomi del libro di ricette.
- `mode`, `group` e `owner` assegnano al file di configurazione gli stessi diritti di accesso dell'applicazione.
- La sezione `variables` imposta la variabile `@redis` utilizzata nel modello sul valore attributo `[:redis]` dell'applicazione.

  I valori attributo `[:redis]` sono impostati utilizzando un JSON personalizzato, come descritto più avanti; non è uno degli attributi delle app standard.
- La direttiva `not_if` garantisce che la ricetta non generi un file di configurazione se ne esiste già uno.

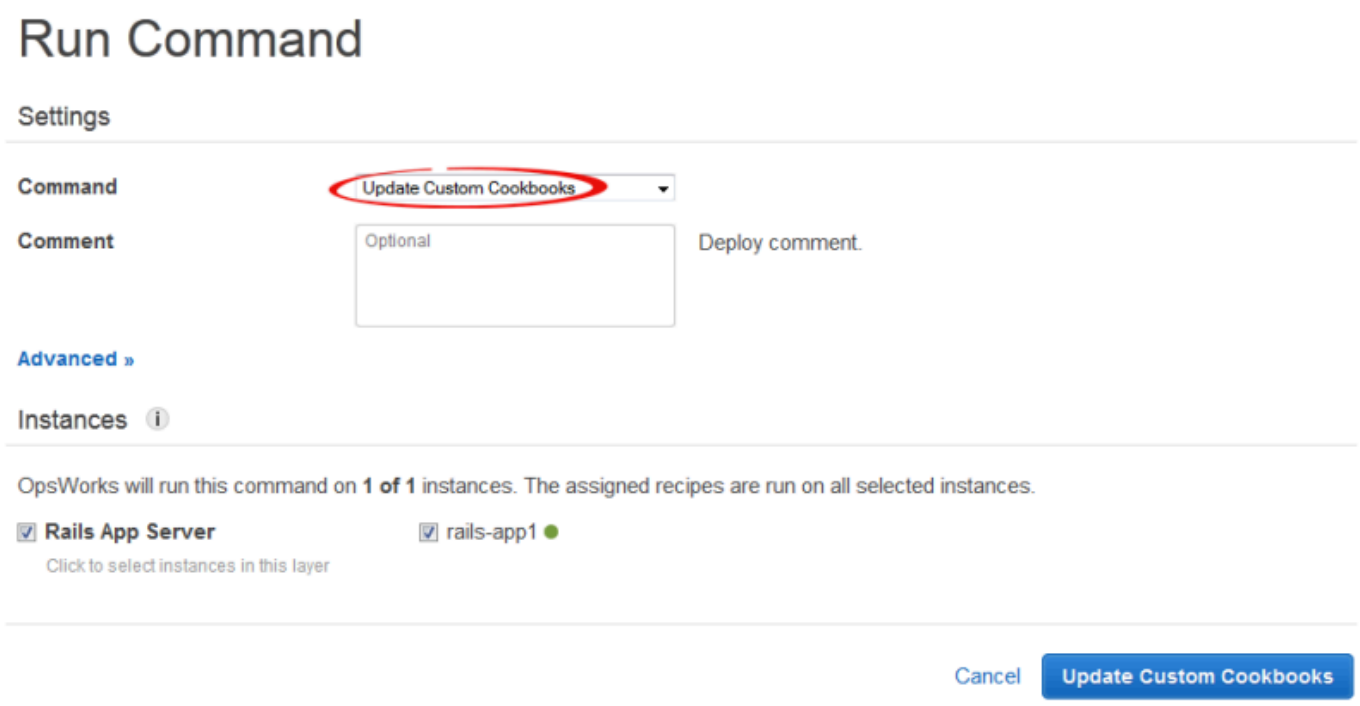
Dopo aver creato il libro di ricette, è necessario distribuirlo per ogni istanza della relativa cache. Questa operazione non esegue la ricetta, ma si limita a installare il nuovo libro di ricette sulle istanze dello stack. Una ricetta viene eseguita in genere assegnandola all'evento del ciclo di vita di un livello, come descritto più avanti.

Per distribuire il libro di ricette personalizzato

1. Nella pagina AWS OpsWorks Stacks Stack, fai clic su Impostazioni dello stack, quindi su Modifica.
2. Nella sezione Configuration Management (Gestione configurazione), impostare Use custom Chef cookbooks (Usa libri di ricette Chef personalizzati) su Yes (Sì), inserire le informazioni sul repository dei libri di ricette e fare clic su Save (Salva) per aggiornare la configurazione dello stack.



3. Nella pagina Stack, fare clic su Run Command (Esegui comando), selezionare il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati) e fare clic su Update Custom Cookbooks (Aggiorna libri di ricette personalizzati) per installare il nuovo libro di ricette nelle cache relative ai libri di ricette nelle istanze.



Per modificare il libro di ricette, basta eseguire di nuovo Update Custom Cookbooks (Aggiorna libri di ricette personalizzati) per installare la versione aggiornata. Per ulteriori informazioni su questa procedura, consulta Installazione di libri di ricette personalizzati.

Fase 4: Assegna la ricetta a un evento LifeCycle

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

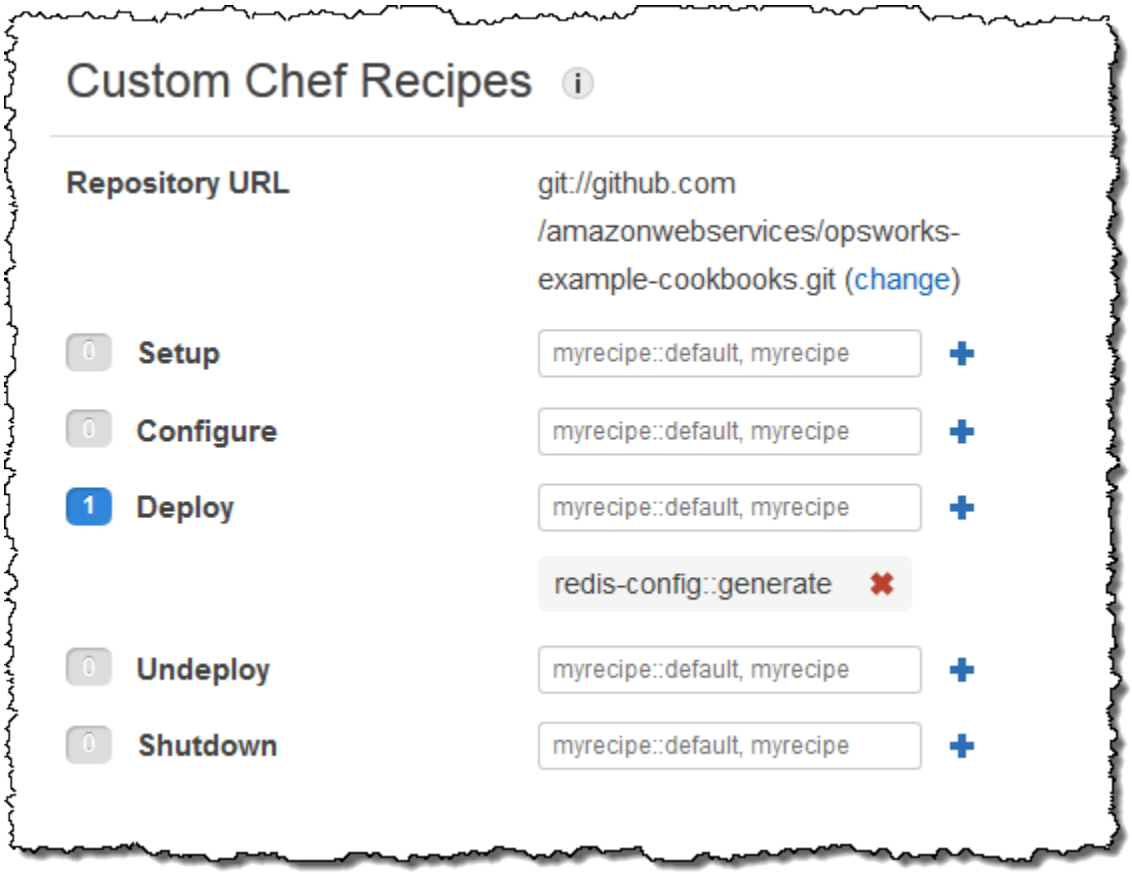
Puoi eseguire ricette personalizzate manualmente, ma l'approccio migliore di solito è quello di far eseguire le ricette automaticamente da AWS OpsWorks Stacks. Ogni livello dispone di una serie di ricette integrate assegnate a ciascuno dei cinque eventi del ciclo di vita: Setup, Configure, Deploy, Undeploy e Shutdown. Ogni volta che si verifica un evento per un'istanza, AWS OpsWorks Stacks esegue le ricette associate per ciascuno dei livelli dell'istanza, che si occupano dell'attività corrispondente. Ad esempio, quando un'istanza termina l'avvio, Stacks attiva un evento Setup.

AWS OpsWorks Tale evento esegue le ricette di configurazione del livello associate, che in genere gestiscono attività quali l'installazione e la configurazione di pacchetti.

È possibile fare in modo che AWS OpsWorks Stacks esegua una ricetta personalizzata sulle istanze di un layer assegnando la ricetta all'evento del ciclo di vita appropriato. Per questo esempio, è necessario assegnare la `generate.rb` ricetta all'evento Deploy del layer Rails App Server. AWS OpsWorks Stacks la eseguirà quindi sulle istanze del layer durante l'avvio, al termine delle ricette di installazione e ogni volta che distribuisci un'app. Per ulteriori informazioni, consulta Esecuzione automatica di ricette.

Per assegnare una ricetta all'evento Deploy del layer Rails App Server

1. Nella pagina AWS OpsWorks Stacks Layers, per Rails App Server, fai clic su Recipes, quindi su Modifica..
2. In Custom Chef Recipes (Ricette Chef personalizzate), aggiungere il nome completo della ricetta all'evento Distribuzione e fare clic su +. Il nome completo della ricetta segue il formato *`cookbookname`*`::`*`recipename`* , dove *`recipename`* non include l'estensione `.rb`. Per questo esempio, il nome completo è `redis-config::generate`. Quindi, fare clic su Save (Salva) per aggiornare la configurazione del livello.



Fase 5: aggiungere informazioni di accesso al JSON di configurazione dello stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

La ricetta `generate.rb` dipende da un paio di attributi JSON di configurazione e distribuzione dello stack che rappresentano il nome host e la porta del server Redis. Sebbene questi attributi facciano parte del `[:deploy]` namespace standard, non vengono definiti automaticamente da Stacks. AWS OpsWorks È possibile invece definire gli attributi e i relativi valori aggiungendo un oggetto JSON personalizzato allo stack. Di seguito è mostrato il JSON personalizzato per questo esempio.

Per aggiungere informazioni al JSON di configurazione e distribuzione dello stack

1. Nella pagina AWS OpsWorks Stacks Stack, fai clic su Impostazioni dello stack, quindi su Modifica.
2. Nella sezione Configuration Management (Gestione configurazione) aggiungere le informazioni di accesso al riquadro Custom Chef JSON (JSON Chef personalizzato). Quest'ultimo dovrebbe essere simile all'esempio seguente, con queste modifiche:

   - Sostituire `elasticache_redis_example` con il nome breve dell'app.
   - Sostituisci i `port` valori `host` and con i valori per l'istanza del server ElastiCache Redis in cui hai creato. Fase 1: Creare un cluster ElastiCache Redis

```
{
  "deploy": {
     "elasticache_redis_example": {
       "redis": {
         "host": "mycluster.XXXXXXXXX.amazonaws.com",
         "port": "6379"
       }
     }
```

```
  }
}
```



Il vantaggio di questo approccio è che puoi modificare la porta o il valore dell'host in qualsiasi momento senza toccare il tuo ricettario personalizzato. AWS OpsWorks Stacks unisce il codice JSON personalizzato al JSON integrato e lo installa sulle istanze dello stack per tutti gli eventi del ciclo di vita successivi. Le applicazioni possono quindi accedere ai valori degli attributi utilizzando la sintassi di nodo Chef, come descritto in Fase 3: creazione e distribuzione di un libro di ricette personalizzato. Alla successiva distribuzione di un'app, AWS OpsWorks Stacks installerà un JSON di configurazione e distribuzione dello stack che contiene le nuove definizioni e `generate.rb` creerà un file di configurazione con i valori di porta e host aggiornati.

> **Note**
>
> `[:deploy]` include automaticamente un attributo per ogni applicazione distribuita, perciò `[:deploy][elasticache_redis_example]` è già presente nel JSON della configurazione e dello stack. Tuttavia, `[:deploy][elasticache_redis_example]` non include un `[:redis]` attributo, la sua definizione con JSON personalizzato indica a Stacks di aggiungere tali attributi. AWS OpsWorks `[:deploy][elasticache_redis_example]` È inoltre possibile utilizzare JSON personalizzati per sovrascrivere gli attributi esistenti. Per ulteriori informazioni, consulta Sostituzione degli attributi.

Fase 6: distribuzione ed esecuzione dell'app

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questo esempio presuppone che tu abbia già l'applicazione Ruby on Rails che utilizza Redis. Per accedere al file di configurazione, puoi aggiungere il pacchetto gem `redis` al tuo Gemfile e creare un inizializzatore Rails in `config/initializers/redis.rb` come illustrato di seguito:

```
REDIS_CONFIG = YAML::load_file(Rails.root.join('config', 'redis.yml'))
$redis = Redis.new(:host => REDIS_CONFIG['host'], :port => REDIS_CONFIG['port'])
```

Quindi crea un'app per rappresentare la tua applicazione e distribuiscila nelle istanze del livello Rails App Server, che aggiorna il codice dell'applicazione e viene eseguita `generate.rb` per generare il file di configurazione. Quando esegui l'applicazione, utilizzerà l'istanza ElastiCache Redis come archivio chiave-valore in memoria.

Utilizzo di un bucket Amazon S3

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le applicazioni utilizzano spesso un bucket Amazon Simple Storage Service (Amazon S3) per memorizzare elementi di grandi dimensioni come immagini o altri file multimediali. Sebbene AWS

OpsWorks Stacks non fornisca supporto integrato per Amazon S3, puoi personalizzare facilmente uno stack per consentire all'applicazione di utilizzare lo storage Amazon S3. Questo argomento illustra il processo di base per fornire ad Amazon S3 l'accesso alle applicazioni, utilizzando uno stack Linux con un server di applicazioni PHP come esempio. I principi di base si applicano anche agli stack Windows.

I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Argomenti

- Fase 1: Creazione di un bucket Amazon S3
- Fase 2: Creazione di uno stack di server app PHP
- Fase 3: creazione e distribuzione di un libro di ricette personalizzato
- Passaggio 4: Assegnare le ricette agli eventi LifeCycle
- Fase 5: Aggiunta di informazioni di accesso agli attributi di configurazione e distribuzione dello stack
- Fase 6: Distribuzione ed esecuzione PhotoApp

Fase 1: Creazione di un bucket Amazon S3

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per prima cosa, devi creare un bucket Amazon S3. Puoi farlo direttamente utilizzando la console, l'API o la CLI di Amazon S3, ma un modo più semplice per creare risorse è spesso quello di utilizzare un modello. AWS CloudFormation Il seguente modello crea un bucket Amazon S3 per questo esempio e configura il profilo dell'istanza con un ruolo IAM che garantisce l'accesso illimitato al bucket. Puoi quindi utilizzare un'impostazione del livello per collegare il profilo dell'istanza alle istanze del server applicazioni dello stack, perché l'applicazione possa accedere al bucket, come descritto di

seguito. L'utilità dei profili di istanza non si limita ad Amazon S3; sono utili per integrare una varietà di servizi AWS.

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "AppServerRootRole": {
         "Type": "AWS::IAM::Role",
         "Properties": {
            "AssumeRolePolicyDocument": {
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "ec2.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            },
            "Path": "/"
         }
      },
      "AppServerRolePolicies": {
         "Type": "AWS::IAM::Policy",
         "Properties": {
            "PolicyName": "AppServerS3Perms",
            "PolicyDocument": {
               "Statement": [ {
                  "Effect": "Allow",
                  "Action": "s3:*",
                  "Resource": { "Fn::Join" : ["", [ "arn:aws:s3:::", { "Ref" :
 "AppBucket" } , "/*" ]
                  ] }
               } ]
            },
            "Roles": [ { "Ref": "AppServerRootRole" } ]
         }
      },
      "AppServerInstanceProfile": {
         "Type": "AWS::IAM::InstanceProfile",
         "Properties": {
            "Path": "/",
            "Roles": [ { "Ref": "AppServerRootRole" } ]
```

```
          }
       },
      "AppBucket" : {
       "Type" : "AWS::S3::Bucket"
       }
   },
   "Outputs" : {
       "BucketName" : {
           "Value" : { "Ref" : "AppBucket" }
       },
       "InstanceProfileName" : {
           "Value" : { "Ref" : "AppServerInstanceProfile" }
       }
   }
}
```

Quando avvii il modello, si verificano diverse situazioni:

- La `AWS::S3::Bucket` risorsa crea un bucket Amazon S3.
- La risorsa `AWS::IAM::InstanceProfile` crea un profilo dell'istanza che verrà assegnato alle istanze del server applicazioni.
- La risorsa `AWS::IAM::Role` crea il ruolo del profilo dell'istanza.
- La `AWS::IAM::Policy` risorsa imposta le autorizzazioni del ruolo per consentire l'accesso illimitato ai bucket Amazon S3.
- La sezione `Outputs` visualizza i nomi del bucket e del profilo dell'istanza nella console AWS CloudFormation dopo l'avvio del modello.

  Questi valori ti serviranno per configurare lo stack e l'app.

Per ulteriori informazioni su come creare AWS CloudFormation modelli, consulta Learn Template Basics.

Creazione del bucket Amazon S3

1. Copiare il modello di esempio in un file di testo nel sistema.

   Questo esempio presuppone che il nome del file sia `appserver.template`.
2. Apri la console AWS CloudFormation e scegli Create Stack (Crea stack).
3. Nella casella Stack Name (Nome stack) immettere il nome dello stack.

Questo esempio presuppone che il nome del file sia **`AppServer`**.

4. Scegliere Upload template file (Carica file modello), Browse (Sfoglia), selezionare il file `appserver.template` creato nella fase 1 quindi scegliere Next Step (Fase successiva).
5. Nella pagina Specify Parameters (Specifica parametri) selezionare I acknowledge that this template may create IAM resources (Sono consapevole che questo modello può creare risorse IAM) e quindi scegliere Next Step (Fase successiva) in ogni pagina della procedura guidata, fino a raggiungere la fine. Scegli Create (Crea).
6. Dopo che lo AppServerstack ha raggiunto lo stato CREATE_COMPLETE, selezionalo e scegli la scheda Output.

   Potrebbe essere necessario aggiornare più volte per aggiornare lo stato.
7. Nella scheda Outputs, registrate i valori and per un uso successivo. BucketNameInstanceProfileName

> **Note**
>
> AWS CloudFormation utilizza il termine stack per riferirsi alla raccolta di risorse create da un modello; non è lo stesso di uno AWS OpsWorks stack Stacks.

Fase 2: Creazione di uno stack di server app PHP

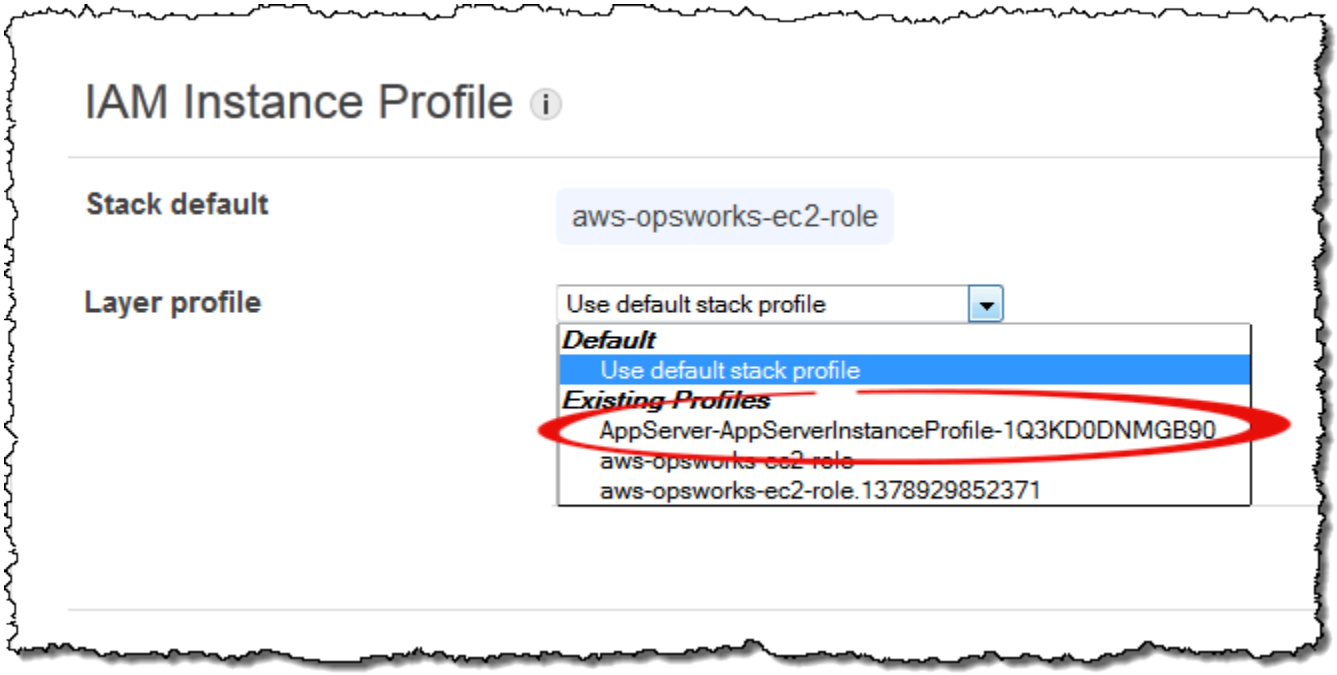
> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Lo stack è composto da due livelli, PHP App Server e MySQL, ciascuno con un'istanza. L'applicazione archivia le foto su un bucket Amazon S3, ma utilizza l'istanza MySQL come archivio dati di back-end per contenere i metadati per ogni foto.

I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Per creare lo stack

1. Crea un nuovo stack, denominato in questo esempio, e aggiungi un livello **`PhotoSite`** PHP App Server. È possibile utilizzare le impostazioni di default per entrambi. Per ulteriori informazioni, consultare Creare un nuovo stack e Creazione di un livello OpsWorks .
2. Nella pagina Livelli, per PHP App Server, scegli Sicurezza, quindi scegli Modifica.
3. Nella sezione Layer Profile, seleziona il nome del profilo dell'istanza che hai registrato in precedenza, dopo aver avviato lo AppServer AWS CloudFormation stack. Sarà qualcosa del genere`AppServer-AppServerInstanceProfile-1Q3KD0DNMGB90`. AWS OpsWorks Stacks assegna questo profilo a tutte le EC2 istanze Amazon del layer, il che concede l'autorizzazione ad accedere al bucket Amazon S3 alle applicazioni in esecuzione sulle istanze del layer.



4. Aggiungi un'istanza al livello PHP App Server e avviala. Per ulteriori informazioni su come aggiungere e avviare istanze, consulta Aggiunta di un'istanza a un livello.
5. Aggiungi un layer MySQL allo stack, aggiungi un'istanza e avviala. È possibile utilizzare le impostazioni predefinite sia per il livello sia per l'istanza. In particolare, l'istanza MySQL non ha bisogno di accedere al bucket Amazon S3, quindi può utilizzare il profilo di istanza Stacks AWS OpsWorks standard, selezionato per impostazione predefinita.

Fase 3: creazione e distribuzione di un libro di ricette personalizzato

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Lo stack non è ancora pronto:

- L'applicazione necessita di alcune informazioni per accedere al server di database MySQL e al bucket Amazon S3, come il nome host del database e il nome del bucket Amazon S3.
- Devi configurare un database nel server di database MySQL e creare una tabella in cui includere i metadati delle foto.

È possibile gestire queste attività manualmente, ma un approccio migliore consiste nell'implementare la ricetta di Chef e fare in modo che AWS OpsWorks Stacks esegua la ricetta automaticamente sulle istanze appropriate. Le ricette Chef sono applicazioni Ruby specializzate che AWS OpsWorks Stacks utilizza per eseguire attività su istanze come l'installazione di pacchetti o la creazione di file di configurazione. Le ricette sono contenute in un libro di ricette, che può includere più ricette e i file correlati, ad esempio i modelli per i file di configurazione. Il ricettario viene inserito in un repository come e deve avere una GitHub struttura di directory standard. Se non disponi già di un repository personalizzato per il libro di ricette, consulta Repository dei libri di ricette per informazioni su come crearne uno.

Per questo esempio, il ricettario è stato implementato per te ed è archiviato in un archivio pubblico. GitHub Il libro di ricette contiene due ricette, `appsetup.rb` e `dbsetup.rb`, e un file di modello, `db-connect.php.erb`.

La `appsetup.rb` ricetta crea un file di configurazione che contiene le informazioni necessarie all'applicazione per accedere al database e al bucket Amazon S3. Si tratta fondamentalmente di una versione leggermente modificata della `appsetup.rb` ricetta descritta in Connessione dell'applicazione al database. La differenza principale consiste nelle variabili che vengono passate al modello, che rappresentano le informazioni di accesso.

I primi quattro attributi definiscono le impostazioni di connessione al database e vengono definiti automaticamente da AWS OpsWorks Stacks quando si crea l'istanza MySQL.

Le differenze tra queste variabili e quelle della ricetta originale sono due:

- Come per la ricetta originale, la variabile `table` rappresenta il nome della tabella di database creata da `dbsetup.rb`ed è impostata sul valore di un attributo definito nel file degli attributi del libro di ricette.

  Tuttavia, l'attributo ha un nome diverso, ovvero `[:photoapp][:dbtable]`.
- La `s3bucket` variabile è specifica di questo esempio ed è impostata sul valore di un attributo che rappresenta il nome del bucket Amazon S3,. `[:photobucket]`

  `[:photobucket]` viene definito tramite codice JSON personalizzato, descritto più avanti. Per ulteriori informazioni sugli attributi, consulta Attributes.

Per ulteriori informazioni sugli attributi, consulta Attributes.

La ricetta `dbsetup.rb` configura una tabella di database in cui includere i metadati di ogni foto. È fondamentalmente una versione leggermente modificata della ricetta `dbsetup.rb` descritta in Configurazione del database. Consultare questo argomento per una descrizione dettagliata.

L'unica differenza tra questo esempio e la ricetta originale è lo schema del database, che ha tre colonne che contengono l'ID, l'URL e la didascalia di ogni foto archiviata nel bucket Amazon S3.

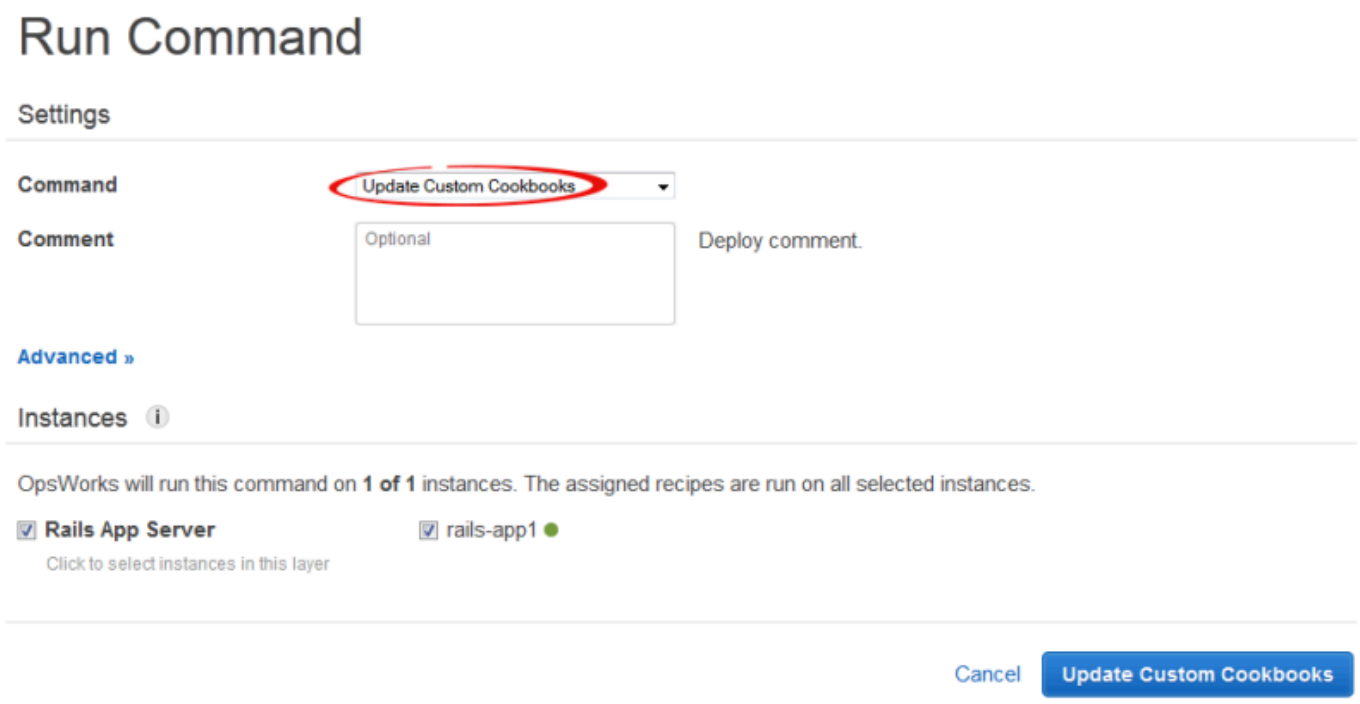
Le ricette sono già implementate, quindi tutto ciò che devi fare è distribuire il ricettario di photoapp nella cache dei libri di cucina di ogni istanza. AWS OpsWorks Stacks esegue quindi le ricette memorizzate nella cache quando si verifica l'evento del ciclo di vita appropriato, come descritto più avanti.

Per distribuire il libro di ricette photoapp

1. Nella pagina AWS OpsWorks Stacks Stack, scegli Stack Settings, quindi scegli Modifica.
2. Nella sezione Configuration Management (Gestione configurazione):
   - Impostare Use custom Chef Cookbooks (Utilizza i libri di ricette di Chef personalizzati) su Yes (Sì).
   - Impostare Repository type (Tipo di repository) su Git.

- Impostare Repository URL (URL repository) su **`git://github.com/amazonwebservices/opsworks-example-cookbooks.git`**.

3. Nella pagina Stack scegliere Run Command (Esegui comando), selezionare il comando dello stack Update Custom Cookbooks (Aggiorna libri di ricette personalizzati) quindi scegliere Update Custom Cookbooks (Aggiorna libri di ricette personalizzati) per installare il nuovo libro di ricette nelle cache dei libri di ricette dell'istanza.



Passaggio 4: Assegnare le ricette agli eventi LifeCycle

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

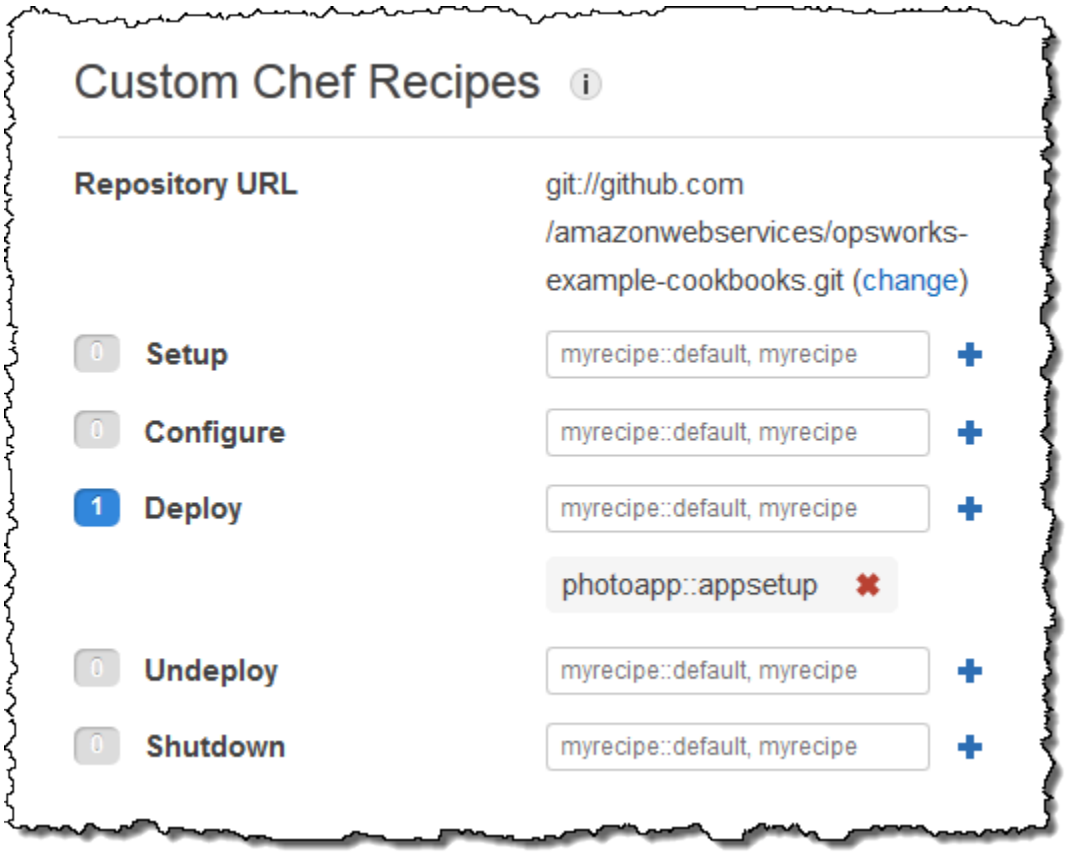
Puoi eseguire ricette personalizzate manualmente, ma l'approccio migliore di solito è quello di far eseguire le ricette automaticamente da AWS OpsWorks Stacks. Ogni livello ha una serie di ricette integrate assegnate a ciascuno dei cinque eventi del ciclo di vita: Setup, Configure, Deploy, Undeploy

e Shutdown—. Ogni volta che si verifica un evento su un'istanza, AWS OpsWorks Stacks esegue le ricette associate per ciascuno dei livelli dell'istanza, che gestiscono le attività richieste. Ad esempio, al termine dell'avvio di un'istanza, AWS OpsWorks Stacks attiva un evento Setup per eseguire le ricette di installazione, che in genere gestiscono attività come l'installazione e la configurazione dei pacchetti.

È possibile fare in modo che AWS OpsWorks Stacks esegua ricette personalizzate sulle istanze di un layer assegnando ogni ricetta all'evento del ciclo di vita appropriato. AWS OpsWorks Stacks eseguirà tutte le ricette personalizzate al termine delle ricette integrate nel layer. Per questo esempio, `appsetup.rb` assegna all'evento Deploy del livello PHP App Server e all'evento Deploy `dbsetup.rb` del livello MySQL. AWS OpsWorks Stacks eseguirà quindi le ricette sulle istanze del layer associato durante l'avvio, al termine delle ricette di installazione integrate e ogni volta che distribuisci un'app, al termine delle ricette Deploy integrate. Per ulteriori informazioni, consulta Esecuzione automatica di ricette.

Per assegnare ricette personalizzate all'evento Deploy (Distribuzione) del livello

1. Nella pagina AWS OpsWorks Stacks Layers, per PHP App Server scegli Recipes, quindi scegli Modifica.
2. In Custom Chef Recipes (Ricette Chef personalizzate), aggiungere il nome della ricetta all'evento di distribuzione e scegliere +. Il nome deve essere nel formato *`cookbookname`*`::`*`recipename`* di Chef, dove *`recipename`* non include l'estensione `.rb`. Per questo esempio, il nome immesso è `photoapp::appsetup`. Quindi scegliere Save (Salva) per aggiornare la configurazione del livello.



3. Nella pagina Livelli, scegli modifica nella colonna Azioni del layer MySQL.
4. Aggiungere `photoapp::dbsetup` all'evento Deploy (Distribuzione) del livello e salvare la nuova configurazione.

Fase 5: Aggiunta di informazioni di accesso agli attributi di configurazione e distribuzione dello stack

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

La `appsetup.rb` ricetta dipende dai dati degli attributi di configurazione e distribuzione dello stack AWS OpsWorks Stacks, che sono installati su ogni istanza e contengono informazioni dettagliate

sullo stack e sulle app distribuite. Gli attributi `deploy` dell'oggetto hanno la struttura seguente, visualizzata per praticità come JSON:

```
{
   ...
  "deploy": {
    "app1": {
      "application" : "short_name",
      ...
    }
    "app2": {
      ...
    }
    ...
  }
}
```

Il nodo di distribuzione contiene un attributo per ogni app distribuita, cui è assegnato il nome breve dell'app. Ogni attributo dell'app contiene un set di attributi che definiscono la configurazione dell'app, ad esempio la root del documento e il tipo di app. Per un elenco degli attributi `deploy`, consulta Attributi deploy. Puoi rappresentare i valori degli attributi di configurazione e distribuzione dello stack nelle ricette utilizzando la sintassi degli attributi di Chef. Ad esempio, `[:deploy][:app1][:application]` rappresenta il nome breve dell'app app1.

Le ricette personalizzate dipendono da diversi attributi di configurazione e distribuzione dello stack che rappresentano il database e le informazioni di accesso ad Amazon S3:

- Gli attributi di connessione al database, ad esempio`[:deploy][:database][:host]`, sono definiti da AWS OpsWorks Stacks quando crea il livello MySQL.
- L'attributo del nome di tabella, `[:photoapp][:dbtable]`, viene definito nel file di attributi del libro di ricette personalizzato ed è impostato su `foto`.
- Devi definire l'attributo del nome del bucket, `[:photobucket]`, utilizzando output JSON personalizzato per aggiungere l'attributo agli attributi di configurazione e distribuzione dello stack.

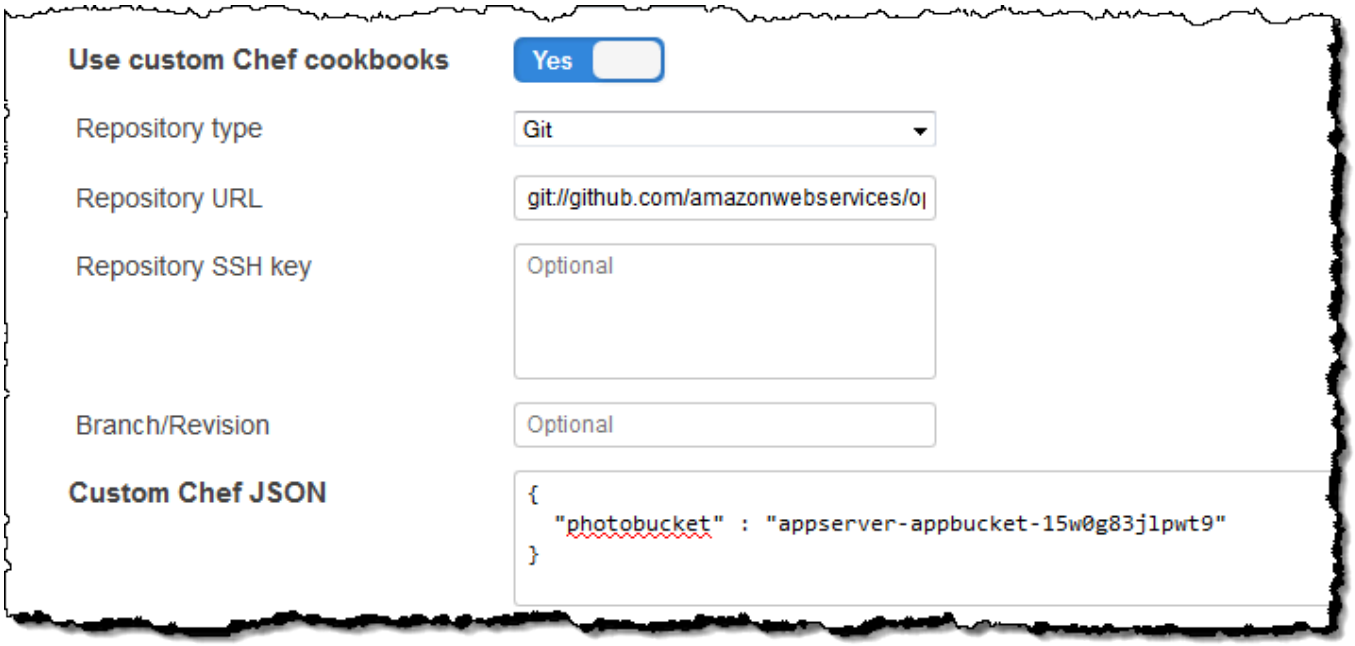
Per definire l'attributo del nome del bucket Amazon S3

1. Nella pagina AWS OpsWorks Stacks Stack, scegli Stack Settings, quindi Modifica.

2. Nella sezione Configuration Management (Gestione configurazione) aggiungere le informazioni di accesso al riquadro Custom Chef JSON (JSON Chef personalizzato). Deve avere un aspetto simile al seguente:

```
{
  "photobucket" : "yourbucketname"
}
```

*yourbucketname*Sostituiscilo con il nome del bucket in cui hai registrato. Fase 1: Creazione di un bucket Amazon S3



AWS OpsWorks Stacks unisce il codice JSON personalizzato agli attributi di configurazione e distribuzione dello stack prima di installarli sulle istanze dello stack; `appsetup.rb` può quindi ottenere il nome del bucket dall'attributo. `[:photobucket]` Se vuoi modificare il bucket, non devi modificare la ricetta, ma puoi semplicemente sostituire l'attributo in modo da specificare un nuovo nome di bucket.

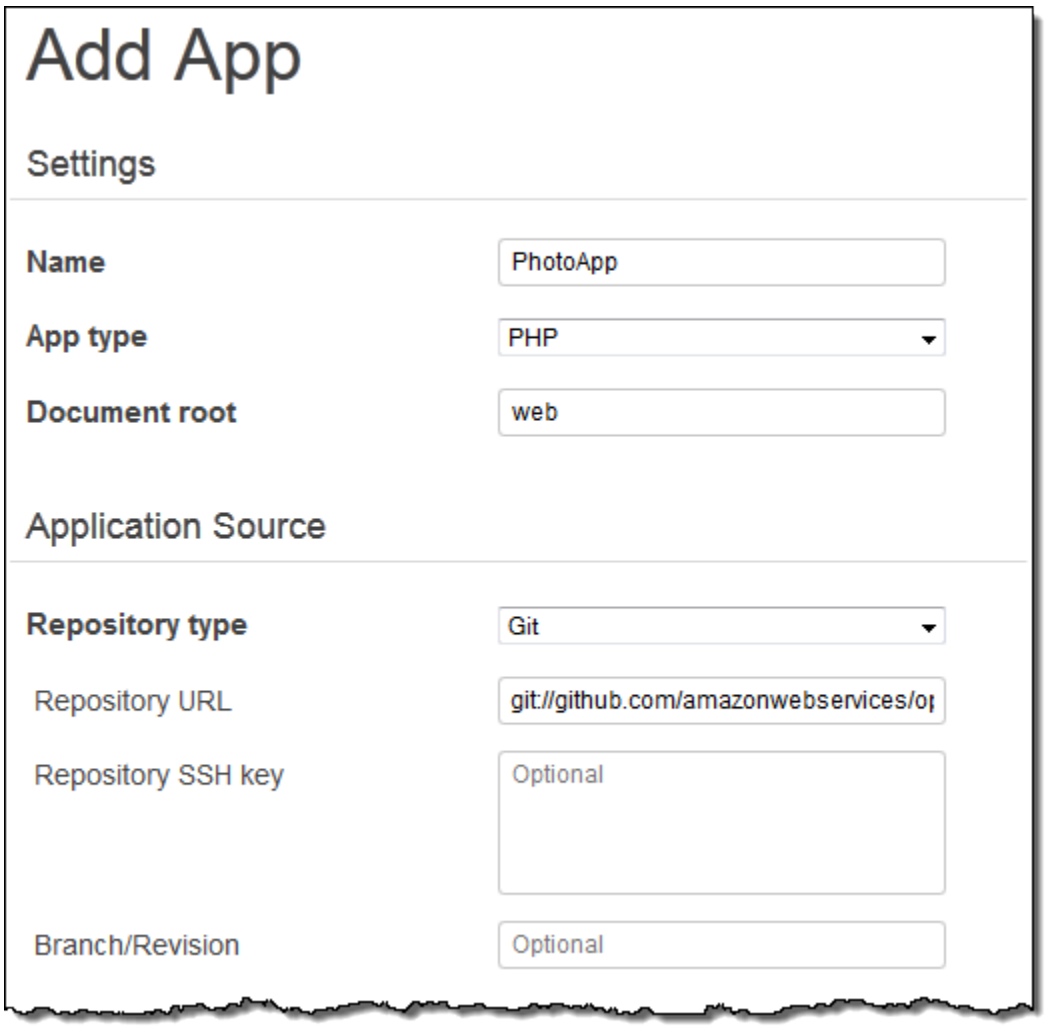
Fase 6: Distribuzione ed esecuzione PhotoApp

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

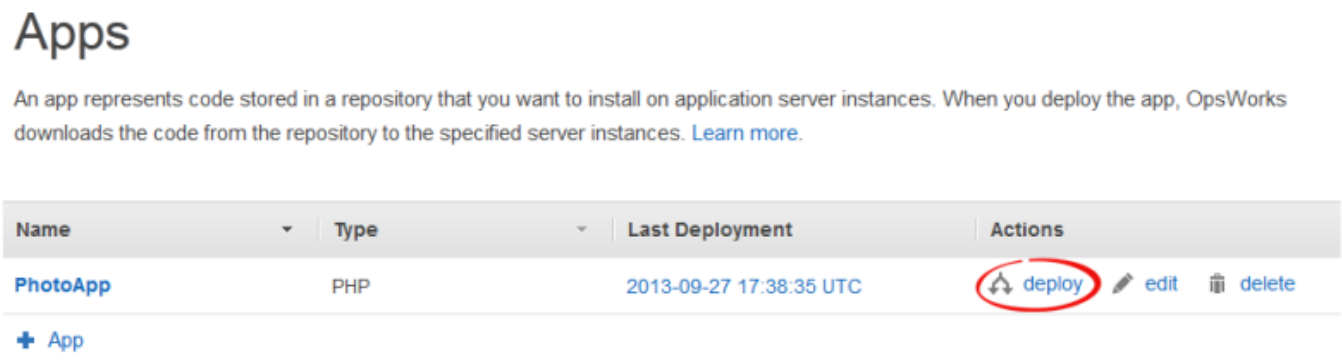
Per questo esempio, l'applicazione è stata implementata anche per te ed è archiviata in un archivio pubblico GitHub . Devi solo aggiungere l'app allo stack, distribuirla nei server applicazioni ed eseguirla.

Per aggiungere l'app allo stack e distribuirla nei server applicazioni

1. Aprire la pagina Apps (App) e scegliere Add an app (Aggiungi un'app).
2. Nella pagina Add App (Aggiungi app) eseguire queste operazioni:
   - Impostare Name (Nome) su **`PhotoApp`**.
   - Impostare App type (Tipo di app) su PHP.
   - Impostare Document root (Root documento) su **`web`**.
   - Impostare Repository type (Tipo di repository) su Git.
   - Impostare Repository URL (URL repository) su **`git://github.com/awslabs/opsworks-demo-php-photo-share-app.git`**.
   - Scegliere Add App (Aggiungi app) per accettare i valori predefiniti per le altre impostazioni.

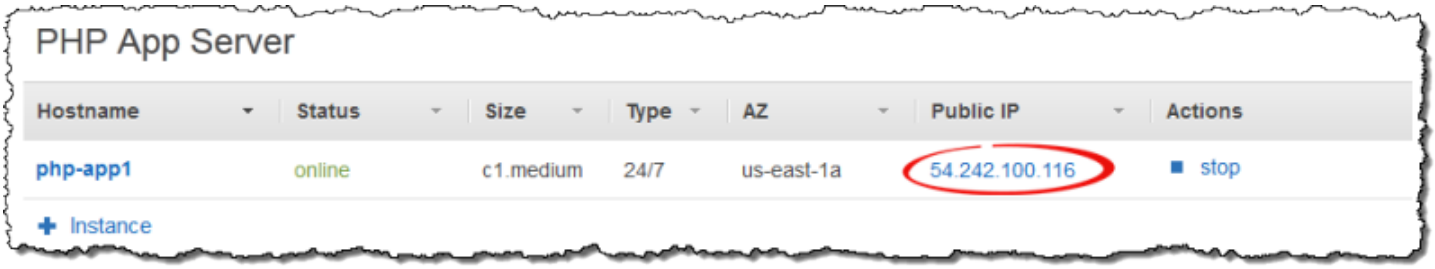


3. Nella pagina App, scegli distribuisci nella colonna Azioni dell' PhotoApp app.



4. Accettare le impostazioni predefinite e scegliere Deploy (Distribuisci) per distribuire l'app nel server.

Per eseguire PhotoApp, vai alla pagina Istanze e scegli l'indirizzo IP pubblico dell'istanza di PHP App Server.



Verrà visualizzata l'interfaccia utente seguente. Scegli Aggiungi una foto per archiviare una foto nel bucket Amazon S3 e i metadati nell'archivio dati di back-end.



Utilizzo AWS CodePipeline con AWS OpsWorks Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS CodePipelineti consente di creare pipeline di distribuzione continua che tengono traccia delle modifiche al codice da fonti come CodeCommit Amazon Simple Storage Service (Amazon S3) o. GitHub Puoi utilizzarlo CodePipeline per automatizzare il rilascio dei libri di cucina Chef e del codice dell'applicazione su AWS OpsWorks Stacks, sugli stack Chef 11.10, Chef 12 e Chef 12.2. Gli esempi in questa sezione descrivono come creare e utilizzare una semplice pipeline CodePipeline come strumento di distribuzione per il codice da eseguire sui livelli Stacks. AWS OpsWorks

> **Note**
>
> CodePipeline e l'integrazione con AWS OpsWorks Stacks non è supportata per la distribuzione su Chef 11.4 e stack precedenti.

Argomenti

- AWS CodePipeline con AWS OpsWorks pile - Chef 12 Stacks
- AWS CodePipeline con AWS OpsWorks Stacks - Chef 11 Stacks

AWS CodePipeline con AWS OpsWorks pile - Chef 12 Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS CodePipelineti consente di creare pipeline di distribuzione continua che tengono traccia delle modifiche al codice da fonti come CodeCommit Amazon Simple Storage Service (Amazon S3) o. GitHub L'esempio in questo argomento descrive come creare e utilizzare una pipeline semplice CodePipeline come strumento di distribuzione per il codice da eseguire su AWS OpsWorks livelli Stacks. In questo esempio, create una pipeline per una semplice app Node.js, quindi chiedete a AWS OpsWorks Stacks di eseguire l'app su tutte le istanze di un livello in uno stack Chef 12 (in questo caso, una singola istanza).

> **Note**
>
> In questo argomento viene descritto come utilizzare una pipeline per eseguire e aggiornare un'app su uno stack Chef 12. Per ulteriori informazioni su come utilizzare una pipeline per eseguire e aggiornare un'app su uno stack Chef 11.10, consulta AWS CodePipeline con AWS OpsWorks Stacks - Chef 11 Stacks. I contenuti distribuiti ai bucket Amazon S3

potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Argomenti

- Prerequisiti
- Altri scenari supportati
- Passaggio 1: crea uno stack, un layer e un'istanza in AWS OpsWorks Stacks
- Fase 2. Configurazione di stack e livello per l'utilizzo di libri di ricette personalizzati
- Fase 3: Caricamento del codice dell'app in un bucket Amazon S3
- Fase 4: Aggiunta dell'app a AWS OpsWorks Stacks
- Fase 5: Creazione di una pipeline in CodePipeline
- Fase 6: Verifica della distribuzione dell'app in AWS OpsWorks Stacks
- Fase 7 (opzionale): Aggiornamento del codice dell'app per impostare CodePipeline per la ridistribuzione automatica dell'app
- Fase 8 (opzionale). Pulizia delle risorse

Prerequisiti

Prima di iniziare questa procedura guidata, verifica di disporre delle autorizzazioni di amministratore per eseguire tutte le attività seguenti. Potete essere membri di un gruppo a cui è stata applicata la AdministratorAccesspolicy oppure potete far parte di un gruppo che dispone delle autorizzazioni e delle politiche mostrate nella tabella seguente. Come procedura consigliata in materia di sicurezza, è consigliabile appartenere a un gruppo che dispone delle autorizzazioni necessarie per eseguire le seguenti attività, anziché assegnare le autorizzazioni necessarie ai singoli utenti.

Per ulteriori informazioni sulla creazione di un gruppo di sicurezza in IAM e sull'assegnazione delle autorizzazioni al gruppo, consulta Creazione di gruppi di utenti IAM. Per ulteriori informazioni sulla gestione delle autorizzazioni AWS OpsWorks Stacks, consulta Best Practices: Managing Permissions.

Autorizzazioni	Policy consigliata da collegare a un gruppo
Crea e modifica pile, livelli e istanze in Stacks. AWS OpsWorks	AWSOpsWorks_FullAccess

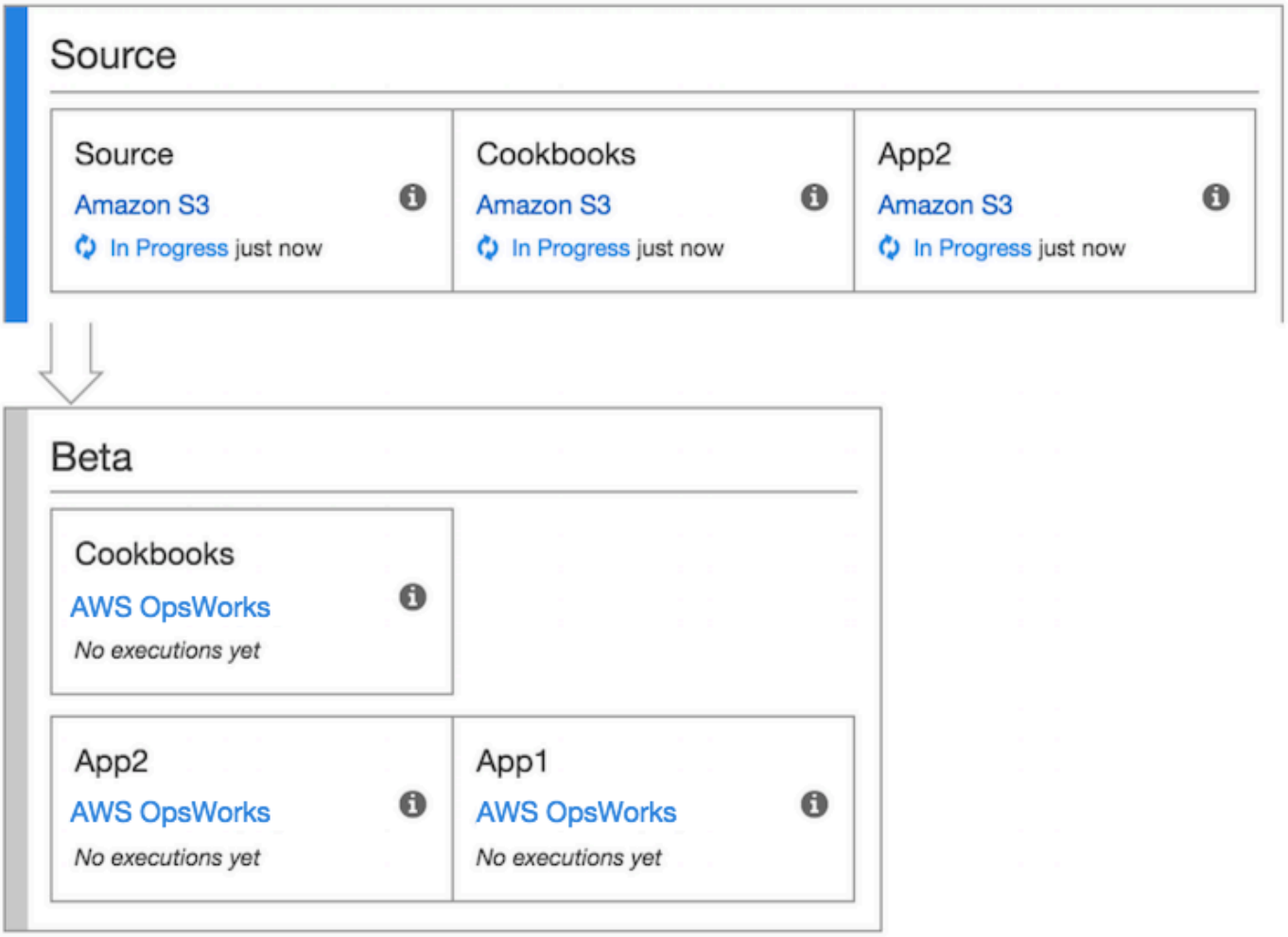
Autorizzazioni	Policy consigliata da collegare a un gruppo
Creare, modificare ed eseguire modelli in AWS CloudFormation.	AmazonCloudFormationFullAccess
Crea, modifica e accedi ai bucket Amazon S3.	Amazon S3 FullAccess
Crea, modifica ed esegui pipeline CodePipeline, in particolare pipeline che utilizzano AWS OpsWorks Stacks come provider.	AWSCodePipeline_FullAccess

È inoltre necessario disporre di una coppia di EC2 chiavi Amazon. Ti verrà richiesto di fornire il nome di questa key pair quando esegui il AWS CloudFormation modello che crea lo stack, il layer e l'istanza di esempio in questa procedura dettagliata. Per ulteriori informazioni su come ottenere una coppia di chiavi nella EC2 console Amazon, consulta Create a Key Pair nella EC2 documentazione di Amazon. La key pair deve trovarsi nella regione Stati Uniti orientali (Virginia settentrionale). Puoi anche utilizzare una coppia di chiavi esistente se ne hai già una in tale regione.

Altri scenari supportati

Questa procedura guidata crea una pipeline semplice che include una fase Source (Origine) e una fase Deploy (Distribuzione). Tuttavia, puoi creare pipeline più complesse che utilizzano AWS OpsWorks Stacks come provider. Di seguito sono illustrati alcuni esempi di pipeline e scenari supportati:

- È possibile modificare una pipeline per aggiungere un libro di ricette Chef alla fase Source (Origine) e una destinazione associata per i libri di ricette aggiornati alla fase Deploy (Distribuzione). In questo caso, si aggiunge un'operazione Deploy (Distribuzione) che attiva l'aggiornamento dei libri di ricette quando vengono apportate modifiche all'origine. Il libro di ricette aggiornato viene distribuito prima dell'app.
- Puoi creare una pipeline complessa, con libri di cucina personalizzati e più app, e distribuirla in uno stack Stacks. AWS OpsWorks La pipeline tiene traccia delle modifiche alle origini sia dell'applicazione che del libro di ricette ed esegue di nuovo la distribuzione in caso di modifiche. Di seguito è illustrato un esempio di una pipeline complessa simile:



Per ulteriori informazioni sull'uso di CodePipeline, consulta la Guida CodePipeline per l'utente.

Passaggio 1: crea uno stack, un layer e un'istanza in AWS OpsWorks Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per utilizzare AWS OpsWorks Stacks come provider di distribuzione per una pipeline, devi prima avere uno stack, un layer e almeno un'istanza nel layer. Sebbene sia possibile creare uno stack in AWS OpsWorks Stacks seguendo le istruzioni in Getting Started with Linux Stacks o Getting Started with Windows Stacks, per risparmiare tempo, questo esempio utilizza un AWS CloudFormation modello per creare uno stack, un layer e un'istanza di Chef 12 basati su Linux. L'istanza creata in base a questo modello esegue Amazon Linux 2016.03 ed è di tipo `c3.large`. Anche se il modello non configura lo stack per l'utilizzo di libri di ricette personalizzati, potrai eseguire questa operazione più avanti nella procedura guidata.

> **Important**
>
> Il AWS CloudFormation modello deve essere archiviato ed eseguito nella stessa regione del bucket Amazon S3 in cui successivamente carichi l'app e nella stessa regione in cui successivamente crei la pipeline. CodePipeline Al momento, CodePipeline supporta il provider AWS OpsWorks Stacks solo nella regione Stati Uniti orientali (Virginia settentrionale) (Virginia settentrionale) (Virginia settentrionale) (us-east-1). Tutte le risorse di questa procedura dettagliata devono essere create nella regione Stati Uniti orientali (Virginia settentrionale).
>
> Se si verifica un errore di creazione dello stack, la causa potrebbe essere il raggiungimento del numero massimo permesso di ruoli IAM per l'account. Può verificarsi un errore nella creazione dello stack anche se l'account non è in grado di avviare istanze di tipo `c3.large`. Ad esempio, se utilizzi il piano AWS gratuito, potresti ricevere un errore del tipo. `Root device type: must be included in EBS` Se il tuo account presenta limitazioni sui tipi di istanze che puoi creare, come le limitazioni imposte dal piano AWS gratuito, prova a modificare il valore del `InstanceType` parametro nel blocco di istanze del modello con un tipo di istanza utilizzabile dal tuo account.

Per creare uno stack, un layer e un'istanza utilizzando AWS CloudFormation

1. Copiate il seguente AWS CloudFormation modello in un nuovo documento di testo semplice. Salvate il file in una posizione comoda sul computer locale e denominatelo NewOpsWorksStack.template o con un altro nome adatto alle vostre esigenze.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Mappings": {
    "Region2Principal": {
```

```
      "us-east-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "us-west-2": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "us-west-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "eu-west-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "ap-southeast-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "ap-northeast-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "ap-northeast-2": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "ap-southeast-2": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "sa-east-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "cn-north-1": {
        "EC2Principal": "ec2.amazonaws.com.cn",
        "OpsWorksPrincipal": "opsworks.amazonaws.com.cn"
      },
      "eu-central-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      }
```

```
    }
 },
 "Parameters": {
   "EC2KeyPairName": {
  "Type": "String",
  "Description": "The name of an existing EC2 key pair that lets you use SSH to
connect to the OpsWorks instance."
 }
 },
 "Resources": {
"CPOpsDeploySecGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription" : "Lets you manage OpsWorks instances to which you deploy
apps with CodePipeline"
  }
},
"CPOpsDeploySecGroupIngressHTTP": {
  "Type": "AWS::EC2::SecurityGroupIngress",
  "Properties" : {
    "IpProtocol" : "tcp",
       "FromPort" : "80",
       "ToPort" : "80",
       "CidrIp" : "0.0.0.0/0",
 "GroupId": {
   "Fn::GetAtt": [
     "CPOpsDeploySecGroup", "GroupId"
   ]
 }
    }
},
"CPOpsDeploySecGroupIngressSSH": {
  "Type": "AWS::EC2::SecurityGroupIngress",
  "Properties" : {
    "IpProtocol" : "tcp",
       "FromPort" : "22",
       "ToPort" : "22",
       "CidrIp" : "0.0.0.0/0",
 "GroupId": {
   "Fn::GetAtt": [
     "CPOpsDeploySecGroup", "GroupId"
   ]
 }
  }
```

```
},
"MyStack": {
     "Type": "AWS::OpsWorks::Stack",
     "Properties": {
       "Name": {
         "Ref": "AWS::StackName"
       },
       "ServiceRoleArn": {
         "Fn::GetAtt": [
           "OpsWorksServiceRole",
           "Arn"
         ]
       },
 "ConfigurationManager" : { "Name": "Chef","Version": "12" },
 "DefaultOs": "Amazon Linux 2016.03",
       "DefaultInstanceProfileArn": {
         "Fn::GetAtt": [
           "OpsWorksInstanceProfile",
           "Arn"
         ]
       },
 "UseCustomCookbooks": "false"
     }
   },
   "MyLayer": {
     "Type": "AWS::OpsWorks::Layer",
     "Properties": {
       "StackId": {
         "Ref": "MyStack"
       },
       "Name": "Node.js App Server",
 "Type": "custom",
       "Shortname": "app1",
 "EnableAutoHealing": "true",
       "AutoAssignElasticIps": "false",
       "AutoAssignPublicIps": "true",
 "CustomSecurityGroupIds": [
   {
     "Fn::GetAtt": [
             "CPOpsDeploySecGroup", "GroupId"
     ]
   }
  ]
     },
```

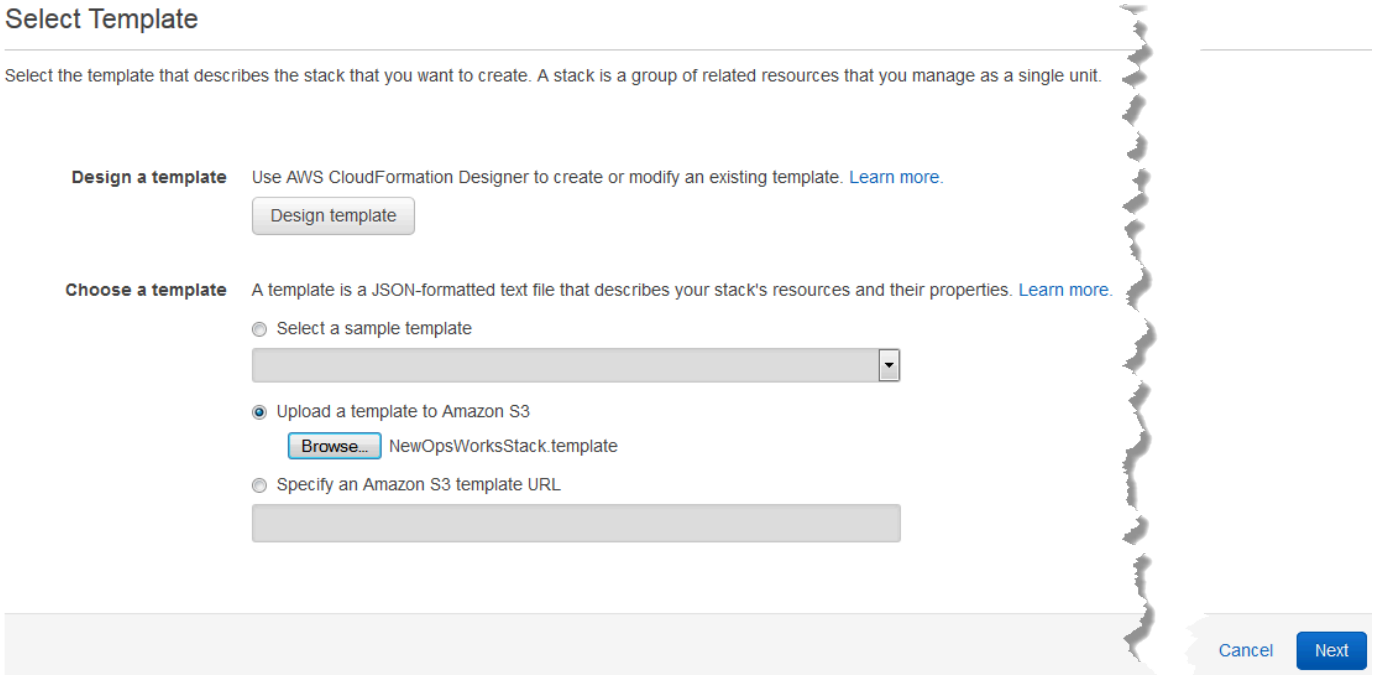
```
      "DependsOn": [
        "MyStack",
        "CPOpsDeploySecGroup"
      ]
    },
    "OpsWorksServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  {
                    "Fn::FindInMap": [
                      "Region2Principal",
                      {
                        "Ref": "AWS::Region"
                      },
                      "OpsWorksPrincipal"
                    ]
                  }
                ]
              },
              "Action": [
                "sts:AssumeRole"
              ]
            }
          ]
        },
        "Path": "/",
        "Policies": [
          {
            "PolicyName": "opsworks-service",
            "PolicyDocument": {
              "Statement": [
                {
                  "Effect": "Allow",
                  "Action": [
                    "ec2:*",
                    "iam:PassRole",
                    "cloudwatch:GetMetricStatistics",
                    "elasticloadbalancing:*"
```

```
                ],
                "Resource": "*"
              }
            ]
          }
        }
      ]
    }
  },
  "OpsWorksInstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
      "Path": "/",
      "Roles": [
        {
          "Ref": "OpsWorksInstanceRole"
        }
      ]
    }
  },
  "OpsWorksInstanceRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": [
                {
                  "Fn::FindInMap": [
                    "Region2Principal",
                    {
                      "Ref": "AWS::Region"
                    },
                    "EC2Principal"
                  ]
                }
              ]
            },
            "Action": [
              "sts:AssumeRole"
            ]
          }
```

```
          ]
        },
        "Path": "/",
  "Policies": [
          {
            "PolicyName": "s3-get",
            "PolicyDocument": {
              "Version": "2012-10-17",
              "Statement": [
                {
                  "Effect": "Allow",
                  "Action": [
                    "s3:GetObject"
                  ],
                  "Resource": "*"
                }
              ]
            }
          }
        ]
      }
    },
    "myinstance": {
      "Type": "AWS::OpsWorks::Instance",
      "Properties": {
        "LayerIds": [
          {
            "Ref": "MyLayer"
          }
        ],
        "StackId": {
          "Ref": "MyStack"
        },
        "InstanceType": "c3.large",
        "SshKeyName": {
    "Ref": "EC2KeyPairName"
  }
      }
    }
  },
  "Outputs": {
    "StackId": {
      "Description": "Stack ID for the newly created AWS OpsWorks stack",
      "Value": {
```

```
        "Ref": "MyStack"
      }
    }
  }
}
```

2. Accedi a AWS Management Console e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformazione.
3. Nella AWS CloudFormation home page, scegli Crea stack.
4. Nella pagina Select Template (Seleziona modello) nell'area Choose a template (Scegliere un modello) scegliere Upload a template to Amazon S3 (Carica un modello in Amazon S3) e quindi Browse (Sfoglia).
5. Vai al AWS CloudFormation modello che hai salvato nel passaggio 1, quindi scegli Apri. Nella pagina Select Template (Seleziona modello), selezionare Next (Avanti).



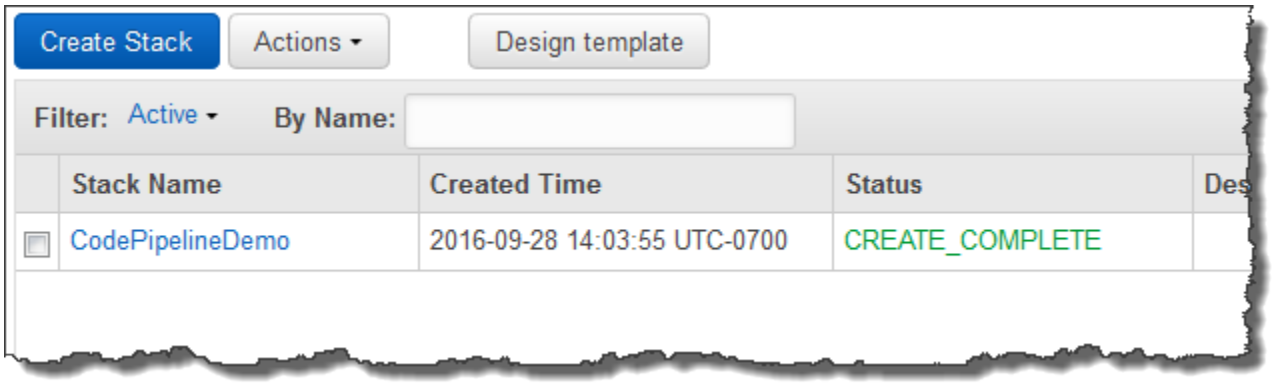
6. Nella pagina Specificare i dettagli, assegna un nome allo stack CodePipelineDemoo a qualsiasi altro nome dello stack univoco per il tuo account. Se si sceglie un nome diverso per lo stack, modificare il nome dello stack in tutta questa procedura guidata.
7. Nell'area Parametri, fornisci il nome di una coppia di EC2 key pair che desideri utilizzare per accedere all'istanza AWS OpsWorks Stacks dopo che è stata creata. Scegli Next (Successivo).
8. Nella pagina Opzioni, scegli Avanti. Le impostazioni in questa pagina non sono necessarie per questa procedura guidata.

9. Il AWS CloudFormation modello utilizzato in questa procedura dettagliata crea ruoli IAM, un profilo di istanza e un'istanza.

   > **Important**
   >
   > Prima di scegliere Create, scegli Costo per stimare gli addebiti che potresti dover sostenere AWS per la creazione di risorse con questo modello.

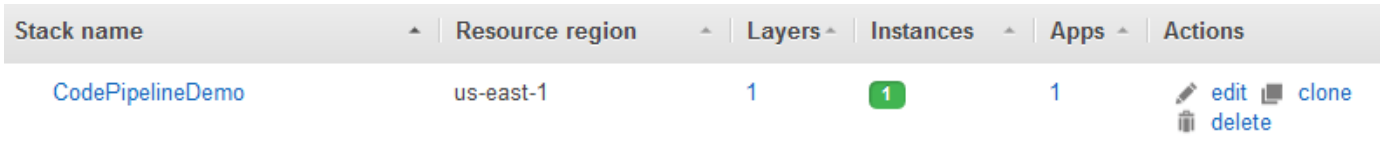
   Se la creazione di risorse IAM è accettabile, seleziona la casella di controllo Riconosco che questo modello potrebbe causare AWS CloudFormation la creazione di risorse IAM, quindi scegli Crea. Se la creazione di risorse IAM non è accettabile, non puoi continuare con questa procedura.

10. Nel AWS CloudFormation pannello di controllo è possibile visualizzare l'avanzamento della creazione dello stack. Prima di continuare con la fase successiva, attendere che venga visualizzato il messaggio CREATE_COMPLETE (CREAZIONE_COMPLETATA) nella colonna Status (Stato).



Per verificare la creazione dello stack in Stacks AWS OpsWorks

1. Apri la AWS OpsWorks console all'indirizzo. https://console.aws.amazon.com/opsworks/
2. Nella dashboard AWS OpsWorks Stacks, visualizza lo stack che hai creato.



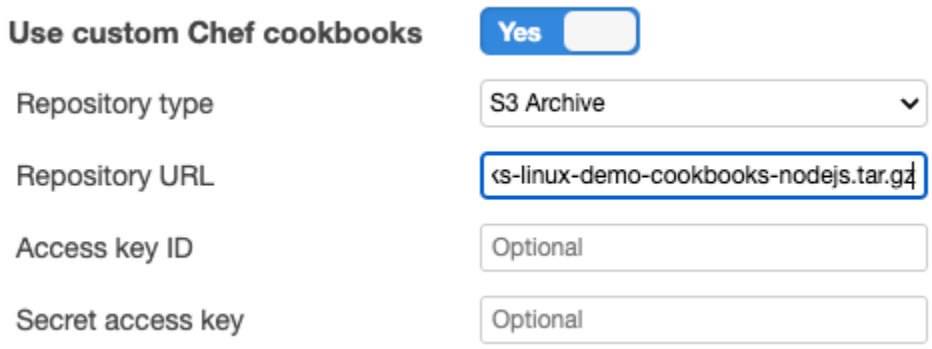
3. Aprire lo stack e visualizzare il livello e l'istanza. Osserva che il layer e l'istanza sono stati creati con i nomi e gli altri metadati forniti nel AWS CloudFormation modello. A questo punto si è pronti per configurare lo stack e il livello per l'utilizzo dei libri di ricette di Chef e le ricette.

Fase 2. Configurazione di stack e livello per l'utilizzo di libri di ricette personalizzati

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Chef 12 stacks in AWS OpsWorks Stacks richiede libri di cucina personalizzati o creati dalla community per creare livelli applicativi personalizzati. Per questa procedura guidata puoi puntare a un archivio contenente un set di libri di ricette di Chef e le ricette di Chef. Queste ricette installano il pacchetto Node.js e le relative dipendenze sull'istanza. Puoi usare altre ricette di Chef per distribuire l'app Node.js che preparerai nella Fase 4: Aggiunta dell'app a AWS OpsWorks Stacks. La ricetta di Chef specificata in questa fase viene eseguita ogni volta che una nuova versione dell'applicazione viene distribuita da CodePipeline.

1. Nella console AWS OpsWorks Stacks, apri lo stack in cui hai creato. Passaggio 1: crea uno stack, un layer e un'istanza in AWS OpsWorks Stacks Scegliere Stack Settings (Impostazioni stack), quindi scegliere Edit (Modifica).
2. Impostare Use custom Chef Cookbooks (Utilizza i libri di ricette di Chef personalizzati) su Yes (Sì). Questa operazione visualizza le impostazioni dei libri di ricette personalizzati correlati.
3. Nell'elenco a discesa Repository type (Tipo di repository), scegliere S3 Archive (Archivio S3). Per funzionare con entrambi CodePipeline e AWS OpsWorks, la fonte del tuo libro di cucina deve essere S3.
4. In Repository URL (URL repository), specificare **`https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-cookbooks-nodejs.tar.gz`**. Le impostazioni dovrebbero essere simili alle seguenti.



5. Scegli Save (Salva).
6. Nel riquadro di navigazione scegli Layers (Livelli).
7. Scegliere Settings (Impostazioni) per il livello creato nella Passaggio 1: crea uno stack, un layer e un'istanza in AWS OpsWorks Stacks.
8. Nella scheda General Settings (Impostazioni generali), assicurarsi che il nome del livello sia Node.js App Server e che il nome breve del livello sia app1. Scegliere Recipes (Ricette).
9. Nella scheda Recipes (Ricette), specificare **`nodejs_demo`** come la ricetta che si desidera eseguire durante l'evento Deploy (Distribuzione) del ciclo di vita. Scegli Save (Salva).
10. Nella scheda Sicurezza, dall'elenco a discesa Gruppi di sicurezza, scegli il gruppo di sicurezza AWS- OpsWorks -Webapp.
11. Scegli Save (Salva).

Fase 3: Caricamento del codice dell'app in un bucket Amazon S3

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

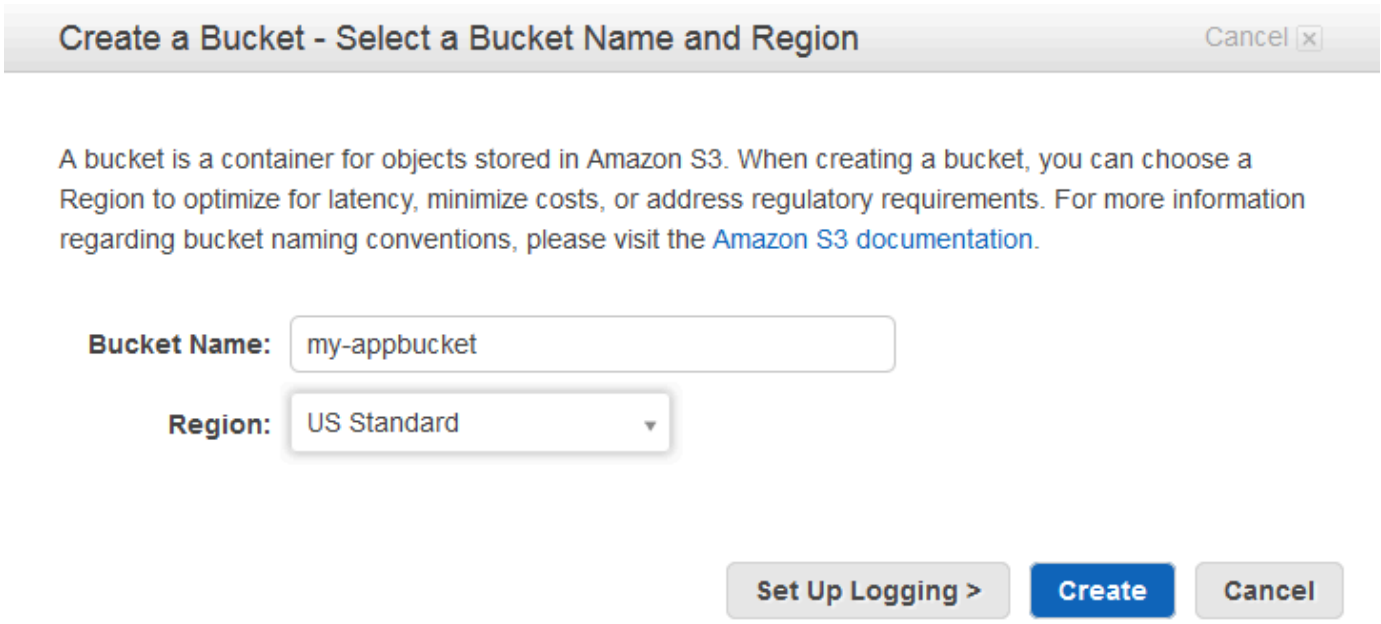
Poiché è necessario fornire un collegamento al repository di codice come parte della configurazione della pipeline, prepara il repository di codice prima di creare la pipeline. Questa procedura dettagliata prevede il caricamento di un'app Node.js in un bucket Amazon S3.

Sebbene sia CodePipeline possibile utilizzare il codice direttamente da GitHub o CodeCommit come sorgente, questa procedura dettagliata dimostra come utilizzare un bucket Amazon S3. In questa procedura dettagliata, carichi l'app Node.js di esempio nel tuo bucket Amazon S3, in modo da poter apportare modifiche all'app. Il bucket Amazon S3 creato in questa fase consente di rilevare le modifiche CodePipeline al codice dell'app e di distribuire automaticamente l'app modificata. Se desideri, puoi utilizzare un bucket esistente. Assicurati che il bucket soddisfi i criteri descritti in Simple Pipeline Walkthrough (Amazon S3 Bucket) nella documentazione. CodePipeline

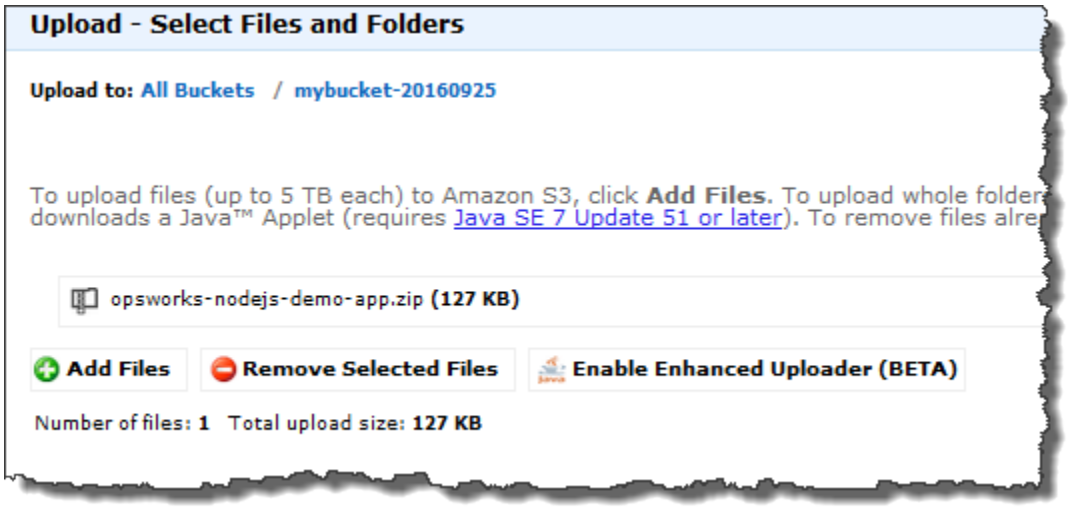
> **Important**
>
> Il bucket Amazon S3 deve trovarsi nella stessa regione in cui creerai la pipeline in un secondo momento. Al momento, CodePipeline supporta il provider AWS OpsWorks Stacks solo nella regione Stati Uniti orientali (Virginia settentrionale) (Virginia settentrionale) (Virginia settentrionale) (us-east-1). Tutte le risorse di questa procedura dettagliata devono essere create nella regione Stati Uniti orientali (Virginia settentrionale). È inoltre necessario che il bucket abbia una versione del codice sorgente, poiché CodePipeline richiede una versione del codice sorgente. Per ulteriori informazioni, consultare Uso della funzione Versioni multiple.

Come caricare l'app in un bucket Amazon S3

1. Scarica il file ZIP dell'esempio AWS OpsWorks Stacks, l'app Node.js, e salvalo in una posizione comoda sul tuo computer locale.
2. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
3. Scegli Crea bucket.
4. Nella pagina Create a Bucket - Select a Bucket Name and Region (Creare un bucket - Selezionare un nome di bucket e una regione), in Bucket Name (Nome bucket), digitare il nome univoco del bucket. I nomi bucket devono essere univoci per tutti AWS gli account, non solo per il tuo. Questa procedura guidata utilizza il nome **`my-appbucket`**, ma è possibile utilizzare `my-appbucket-`*`yearmonthday`* per rendere univoco il nome del bucket. Nell'elenco a discesa Region (Regione), scegliere US Standard (Stati Uniti standard), quindi scegliere Create (Crea). US Standard (Stati Uniti standard) equivale a `us-east-1`.



5. Scegliere il bucket creato nell'elenco Tutti i bucket.
6. Nella pagina del bucket scegliere Upload (Carica).
7. Nella pagina Upload - Select Files and Folders (Carica - Seleziona file e cartelle), scegliere Add Files (Aggiungi file). Cercare il file ZIP salvato nella fase 1, scegliere Open (Apri), quindi scegliere Start Upload (Avvia caricamento).



8. Al termine del caricamento, selezionare il file ZIP nell'elenco di file nel bucket, quindi scegliere Properties (Proprietà).
9. Nel riquadro Properties (Proprietà), copiare il collegamento al file ZIP, quindi prendere nota del collegamento. La parte di collegamento con il nome del bucket e il nome del file ZIP sarà necessaria per creare la pipeline.

Fase 4: Aggiunta dell'app a AWS OpsWorks Stacks

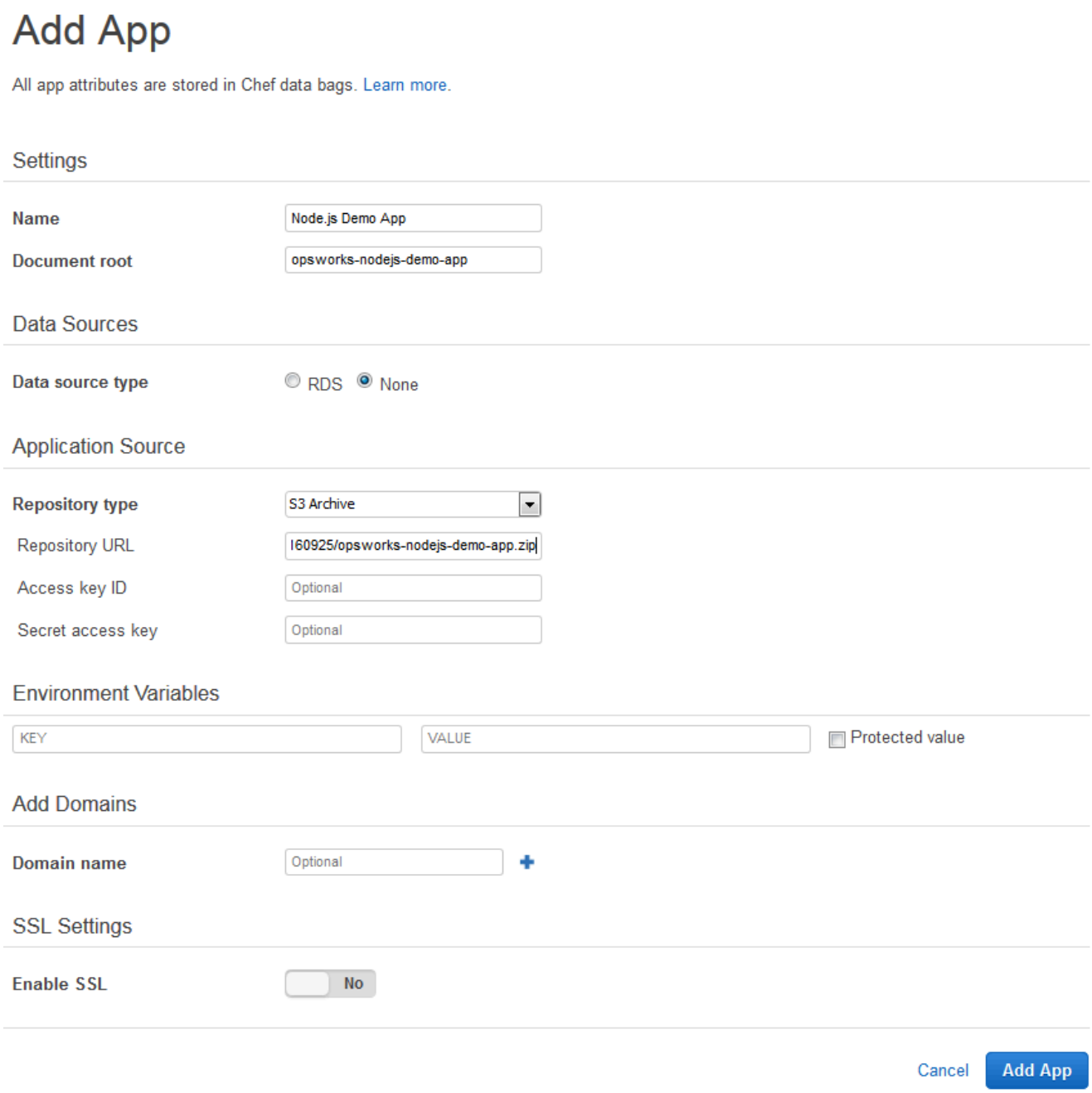
> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Prima di creare una pipeline in CodePipeline, aggiungi l'app di test Node.js a Stacks. AWS OpsWorks Quando crei la pipeline, dovrai selezionare l'app che hai aggiunto a Stacks. AWS OpsWorks

Tieni a portata di mano il link al bucket Amazon S3 del passaggio 9 della procedura precedente. Per completare questa procedura, è necessario il collegamento al bucket in cui hai archiviato l'app di test.

Per aggiungere un'app a Stacks AWS OpsWorks

1. Nella console AWS OpsWorks Stacks CodePipelineDemo, apri e nel riquadro di navigazione scegli App.
2. Scegliere Add app (Aggiungi app).
3. Nella pagina Add app (Aggiungi app), fornire le seguenti informazioni:
   a. Specificare un nome per l'app. Questa procedura guidata utilizza il nome `Node.js Demo App`.
   b. In Data source type (Tipo di origine dati), scegliere None (Nessuno). Questa app non richiede un'origine dati o un database esterno.
   c. Nell'elenco a discesa Repository type (Tipo di repository), scegliere S3 Archive (Archivio S3).
   d. Nella casella di testo Repository URL (URL repository), incollare l'URL copiato nella fase 9 della Fase 3: Caricamento del codice dell'app in un bucket Amazon S3. Il modulo dovrebbe essere simile al seguente:



4. Non è necessario modificare altre impostazioni in questo modulo. Scegliere Add App (Aggiungi app).
5. Quando l'app Node.js Demo App (Applicazione demo Node.js) viene visualizzata nell'elenco della pagina Apps (App), passare alla procedura successiva, Fase 5: Creazione di una pipeline in CodePipeline.

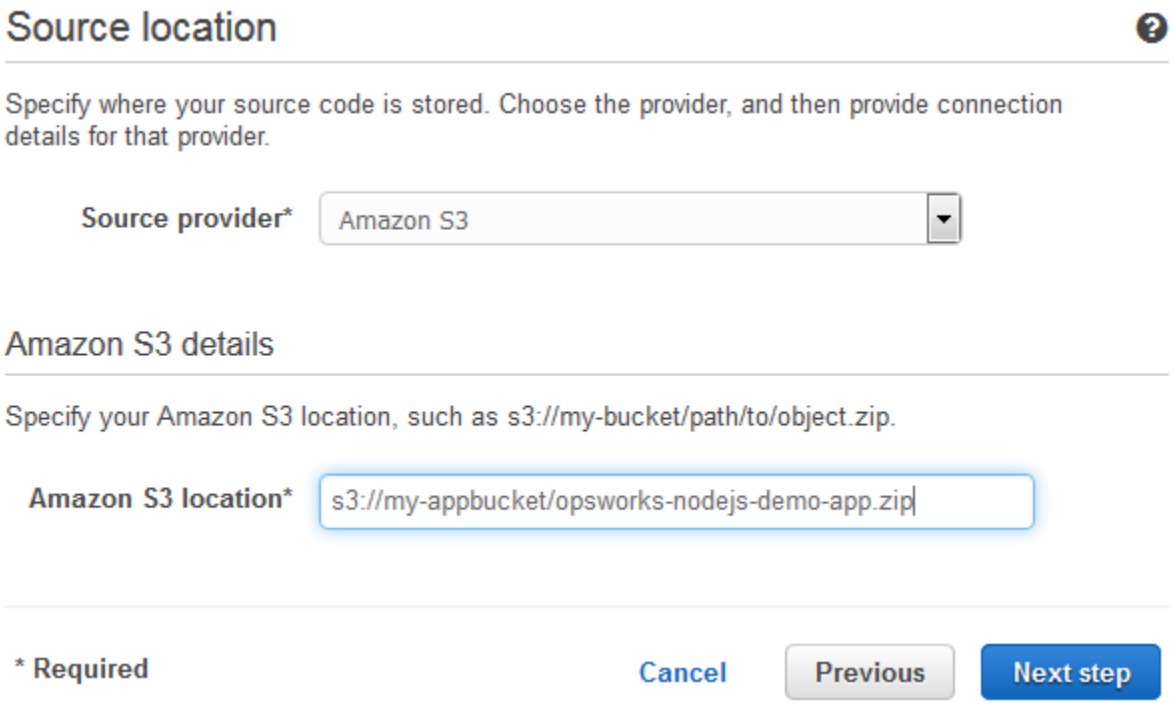
Fase 5: Creazione di una pipeline in CodePipeline

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

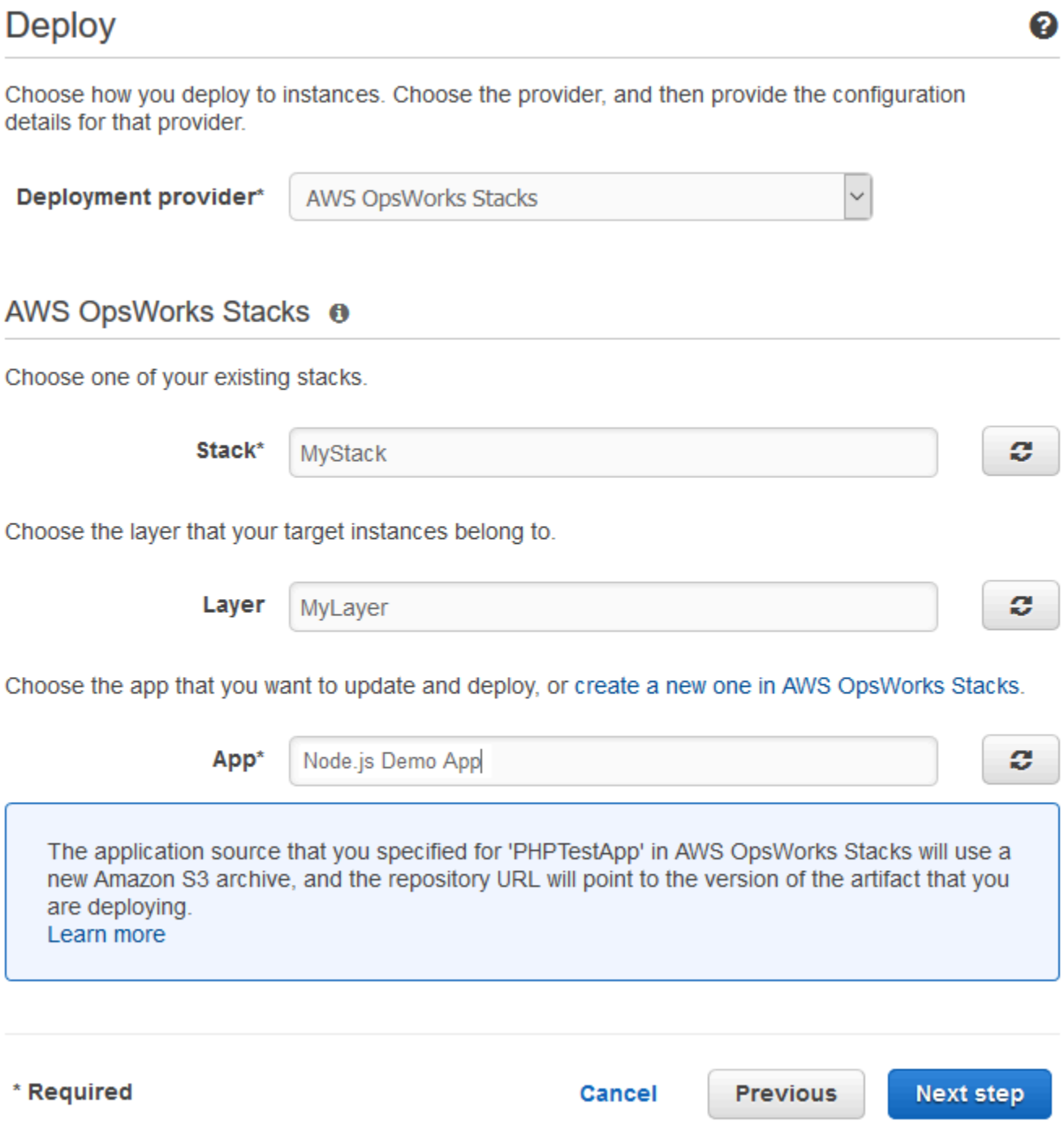
Dopo aver configurato uno stack con un layer e almeno un'istanza configurata in AWS OpsWorks Stacks, crea una pipeline utilizzando AWS OpsWorks Stacks come provider per distribuire app o libri di cucina Chef nelle CodePipeline tue risorse Stacks. AWS OpsWorks

Per creare una pipeline

1. Apri la console all'indirizzo. CodePipeline https://console.aws.amazon.com/codepipeline/
2. Scegliere Create pipeline (Crea pipeline).
3. Nella pagina Getting started with CodePipeline (Nozioni di base su CodePipeline), digitare **`MyOpsWorksPipeline`** o qualsiasi altro nome di pipeline purché sia univoco per l'account in uso, quindi scegliere Next step (Fase successiva).
4. Nella pagina Source Location (Posizione di origine), selezionare Amazon S3 nell'elenco a discesa Source provider (Provider origine).
5. Nell'area dei dettagli di Amazon S3, digita il percorso del bucket Amazon S3, nel formato. **`s3://`*`bucket-name`*`/`*`file name`** Fare riferimento al collegamento annotato nella fase 9 della Fase 3: Caricamento del codice dell'app in un bucket Amazon S3. In questa procedura guidata, il percorso è `s3://my-appbucket/opsworks-nodejs-demo-app.zip`. Seleziona Fase successiva.



6. Nella pagina Build (Crea), scegliere No Build (Nessuna creazione) nell'elenco a discesa, quindi scegliere Next step (Fase successiva).
7. Nella pagina Deploy (Distribuzione), scegliere AWS OpsWorks Stacks come provider di distribuzione.



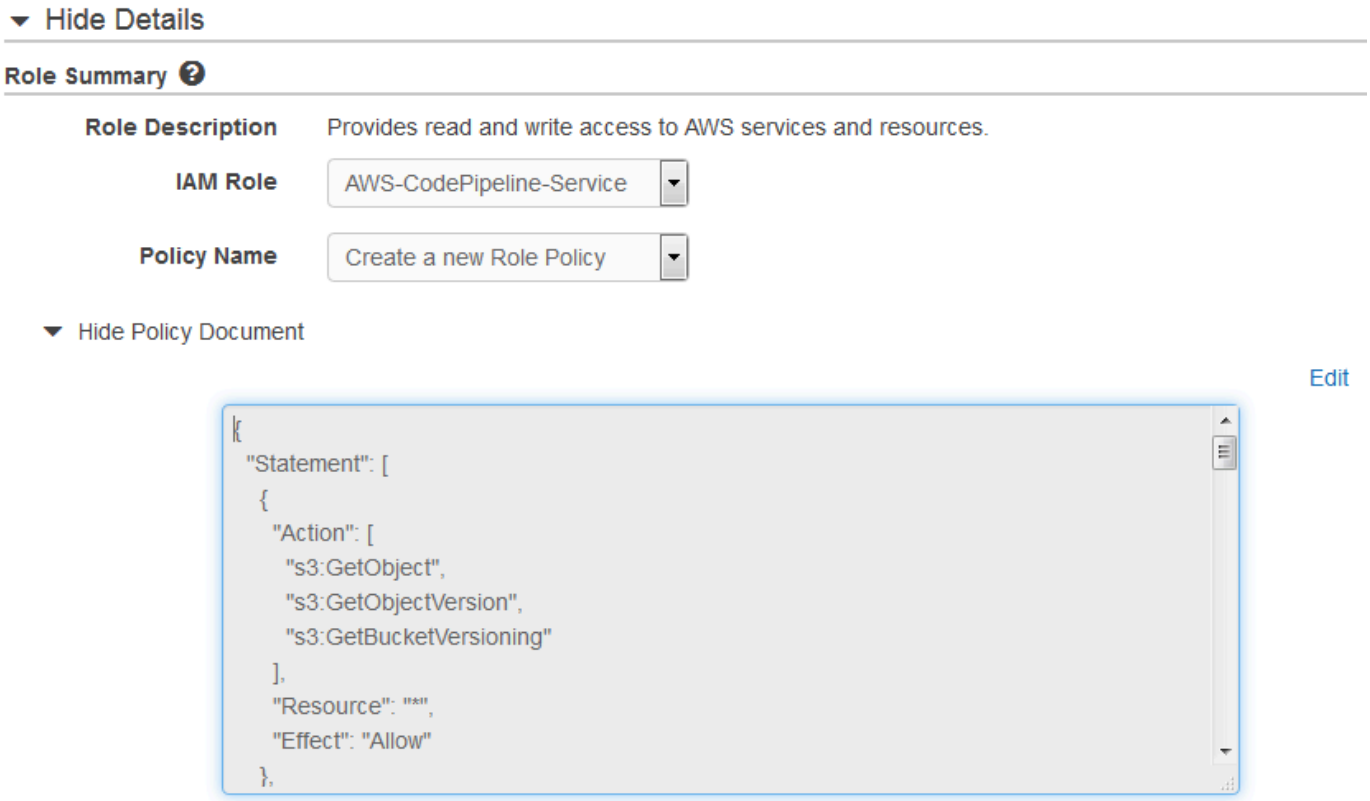
8. Nel campo Stack, digitare `CodePipelineDemo` oppure il nome dello stack creato nella Passaggio 1: crea uno stack, un layer e un'istanza in AWS OpsWorks Stacks.
9. Nel campo Layer (Livello), digitare `Node.js App Server` oppure il nome del livello creato nella Passaggio 1: crea uno stack, un layer e un'istanza in AWS OpsWorks Stacks.

10. Nel campo App, seleziona l'app che hai caricato su Amazon S3Fase 3: Caricamento del codice dell'app in un bucket Amazon S3, quindi scegli Passaggio successivo.
11. Nella pagina AWS Service Role, scegli Create Role.

    Si apre una nuova finestra con una pagina della console IAM che descrive il ruolo che verrà creato per te,`AWS-CodePipeline-Service`. Nell'elenco a discesa Policy name (Nome policy), scegliere Create new policy (Crea una nuova policy). Assicurarsi che il documento di policy abbia il contenuto seguente. Scegliere Edit (Modifica) per modificare il documento delle policy, se necessario.

```
{
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetBucketVersioning"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "opsworks:*",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

    Dopo aver modificato il documento di policy, scegliere Allow (Consenti). Le modifiche verranno visualizzate nella console IAM.



> **Note**
>
> Se la creazione del ruolo fallisce, potrebbe essere perché disponi già di un ruolo IAM denominato AWS- CodePipeline -Service. Se hai utilizzato il ruolo AWS- CodePipeline -Service prima di maggio 2016, il ruolo potrebbe non disporre delle autorizzazioni per utilizzare AWS OpsWorks Stacks come fornitore di distribuzione. In questo caso, è necessario aggiornare l'istruzione della policy come indicato in questa fase. Se viene visualizzato un messaggio di errore, tornare all'inizio di questa fase e scegliere Use existing role (Utilizza un ruolo esistente) anziché Create role (Crea ruolo). Se si utilizza un ruolo esistente, il ruolo deve avere una policy collegata che include le autorizzazioni indicate in questa fase. Per ulteriori informazioni sul ruolo del servizio e sulla relativa istruzione della policy, consulta l'argomento relativo alla modifica di una policy per un ruolo del servizio IAM.

12. Se il processo di creazione del ruolo ha esito positivo, la pagina IAM si chiuderà e verrai reindirizzato alla pagina AWS Service Role. Seleziona Fase successiva.
13. Nella pagina Review your pipeline (Verifica pipeline), verificare le selezioni effettuate nella pagina, quindi scegliere Create pipeline (Crea pipeline).

14. Quando la pipeline è pronta, dovrebbe iniziare a individuare il codice sorgente e a distribuire l'app allo stack automaticamente. Questo processo può richiedere alcuni minuti.

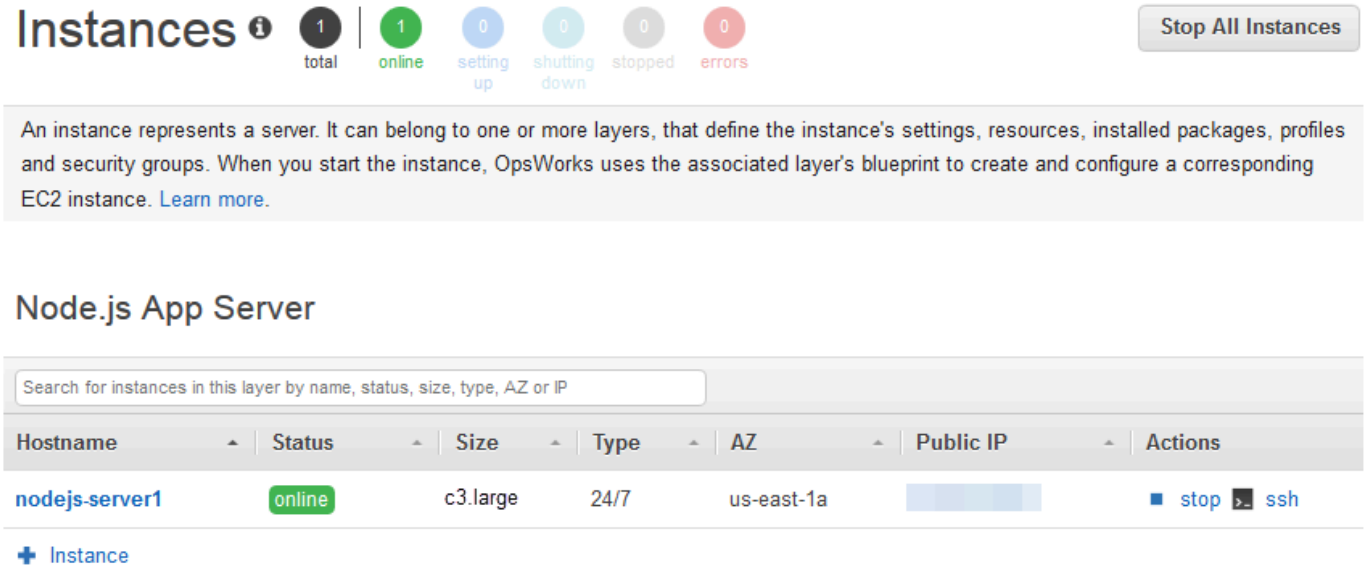
Fase 6: Verifica della distribuzione dell'app in AWS OpsWorks Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

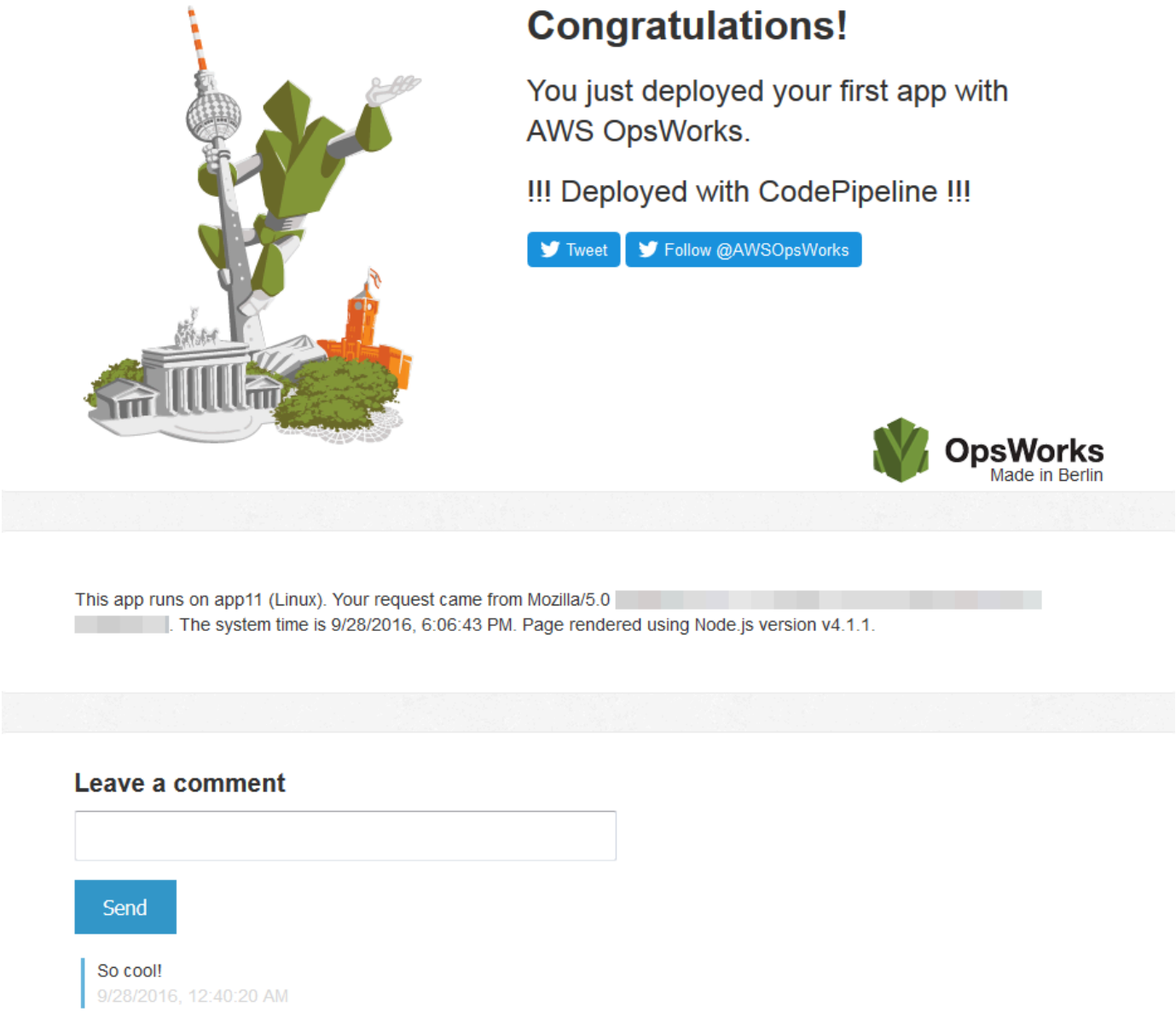
Per verificare di CodePipeline aver distribuito l'app Node.js nel tuo stack, accedi all'istanza che hai creato. Passaggio 1: crea uno stack, un layer e un'istanza in AWS OpsWorks Stacks Dovresti essere in grado di visualizzare e utilizzare l'app Web Node.js.

Per verificare la distribuzione dell'app nella tua AWS OpsWorks istanza Stacks

1. Apri la AWS OpsWorks console all'indirizzo https://console.aws.amazon.com/opsworks/.
2. Nella dashboard AWS OpsWorks Stacks, scegli CodePipelineDemo, quindi scegli Node.js App Server.
3. Nel riquadro di navigazione, scegliere Instances (Istanze), quindi scegliere l'indirizzo IP pubblico dell'istanza creata per visualizzare l'app Web.



L'app verrà visualizzata in una nuova scheda del browser.



Fase 7 (opzionale): Aggiornamento del codice dell'app per impostare CodePipeline per la ridistribuzione automatica dell'app

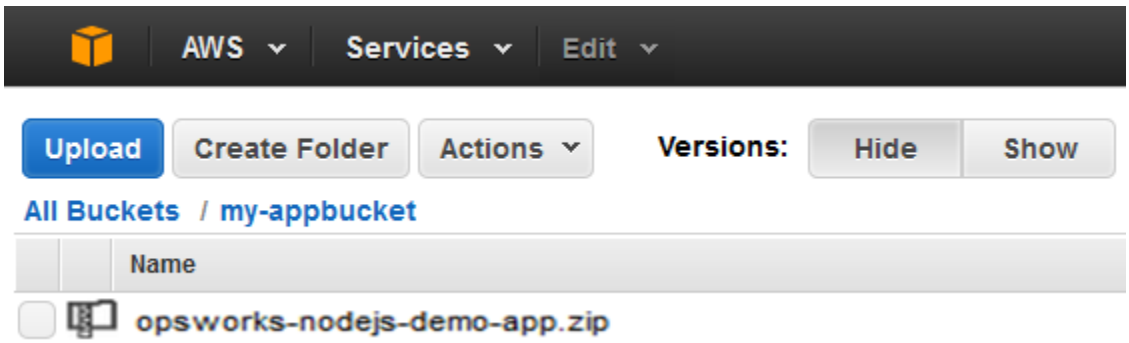
> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Quando apporti modifiche al codice nelle app o nei libri di cucina che hai distribuito utilizzando CodePipeline, gli artefatti aggiornati verranno distribuiti automaticamente sulle istanze di destinazione (in questo caso, CodePipeline su uno stack di stack di destinazione). AWS OpsWorks Questa sezione illustra la ridistribuzione automatica quando aggiorni il codice nell'app Node.js di esempio. Se disponi ancora del codice dell'app per questa procedura guidata archiviato in locale e se non sono state apportate modifiche al codice dall'avvio della procedura guidata, puoi ignorare le fasi 1-4 di questa procedura.

Per modificare il codice dell'app di esempio

1. Accedi a AWS Management Console e apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Aprire il bucket in cui si sta archiviando l'app Node.js di esempio.

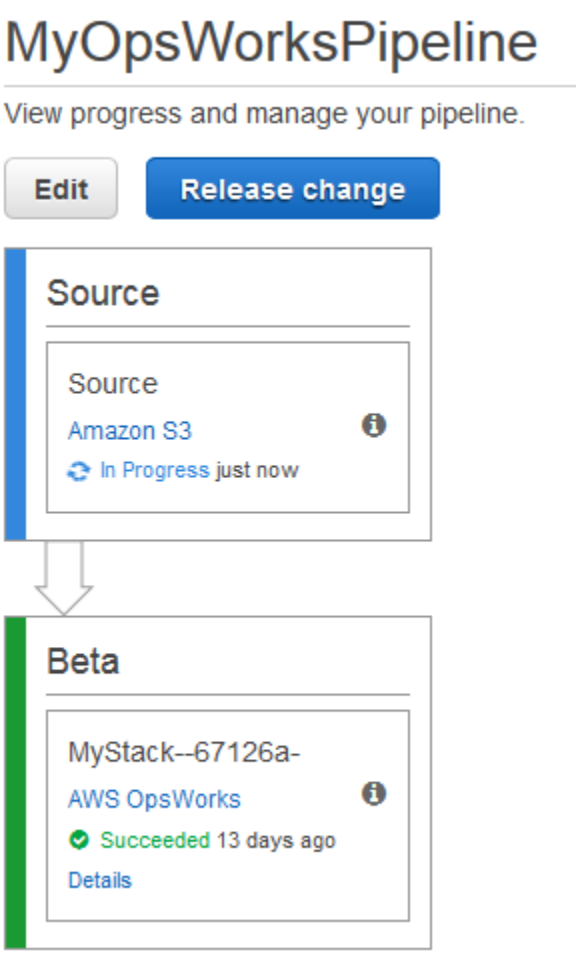


3. Selezionare il file ZIP contenente l'app. Nel menu Actions (Operazioni), scegliere Download (Scarica).
4. Nella finestra di dialogo, aprire il menu contestuale facendo clic con il pulsante destro del mouse, scegliere Download (Scarica), quindi salvare il file ZIP in una posizione comoda. Scegli OK.
5. Estrarre il contenuto del file ZIP in un percorso comodo. Potrebbe essere necessario modificare le autorizzazioni per la cartella estratta, le sottocartelle e i contenuti per permettere la modifica. Nella cartella `opsworks-nodejs-demo-app\views`, aprire il file `header.html` per la modifica.
6. Cercare la frase `You just deployed your first app with`. Sostituire la parola `deployed` con `updated`. Nella riga successiva, modificare `AWS OpsWorks.` in `AWS OpsWorks and AWS CodePipeline.` Modificare esclusivamente il testo.

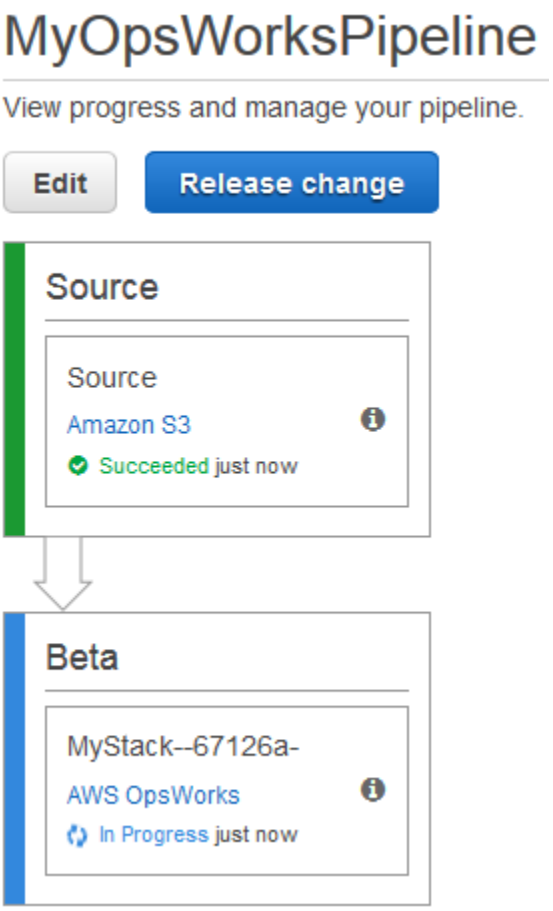
```
<div id="main" role="main">LF
  <div class="container">LF
    <div class="hero-unit">LF
      <div class="robot">LF
        <h1>Congratulations!</h1>LF
        <h2>LF
          You just updated your first app with<br/>LF
          AWS OpsWorks and AWS CodePipeline.LF
        </h2>LF
```

7. Salvare e chiudere il file `header.html`.
8. Comprimere la cartella `opsworks-nodejs-demo-app` e salvare il file ZIP in una posizione comoda. Non modificare il nome del file ZIP.
9. Carica il nuovo file ZIP nel tuo bucket Amazon S3. In questa procedura guidata il nome del bucket è `my-appbucket`.
10. Apri la CodePipeline console e apri la pipeline AWS OpsWorks Stacks (). MyOpsWorksPipeline Scegliere Release Change (Rilascia modifica).

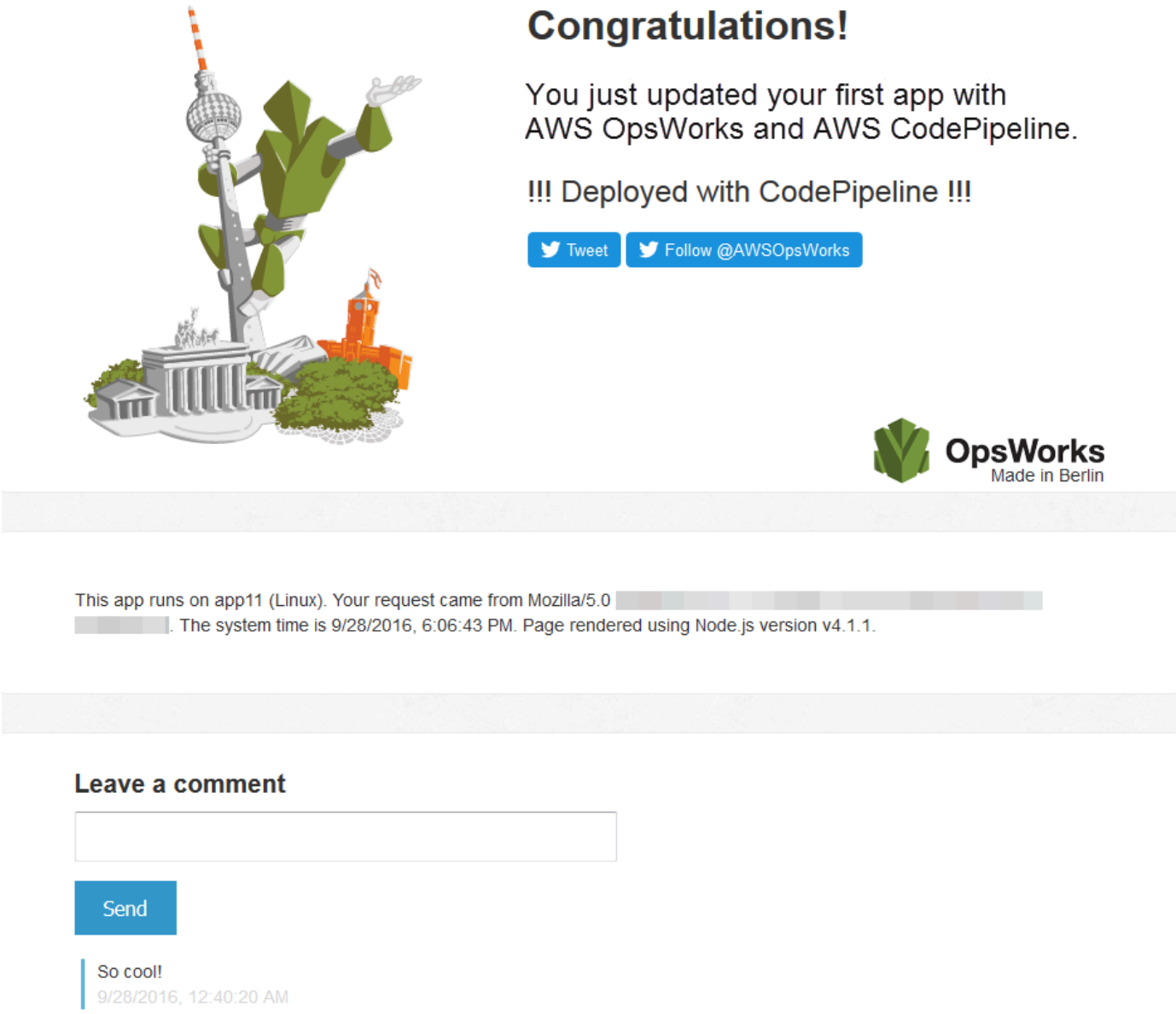
    (Puoi attendere che CodePipeline venga rilevata la modifica del codice dalla versione aggiornata dell'app nel tuo bucket Amazon S3. Per farti risparmiare tempo, questa procedura dettagliata ti spiega semplicemente di scegliere Release Change.)
11. Osserva come CodePipeline scorre lungo le fasi della pipeline. Innanzitutto, CodePipeline rileva le modifiche all'artefatto di origine.



CodePipeline inserisce il codice aggiornato nel tuo stack in Stacks. AWS OpsWorks



12. Quando entrambe le fasi della pipeline sono state completate con successo, aprire lo stack in AWS OpsWorks Stacks.
13. Nella pagina delle proprietà dello stack, scegliere Instances (Istanze).
14. Nella colonna Public IP (IP pubblico), scegliere l'indirizzo IP pubblico dell'istanza per visualizzare il testo dell'app aggiornata.



Fase 8 (opzionale). Pulizia delle risorse

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per evitare addebiti indesiderati AWS sul tuo account, puoi eliminare le AWS risorse utilizzate per questa procedura dettagliata. Queste AWS risorse includono lo AWS OpsWorks stack Stacks, il ruolo e il profilo dell'istanza IAM e la pipeline in cui hai creato. CodePipeline Tuttavia, potresti voler continuare a utilizzare queste AWS risorse man mano che continui a saperne di più su AWS OpsWorks Stacks e. CodePipeline Se desideri mantenere queste risorse, hai completato la procedura guidata.

Per eliminare l'app dallo stack

Poiché non hai creato o applicato l'app come parte del AWS CloudFormation modello, elimina l'app di test Node.js prima di eliminare lo stack in esso contenuto. AWS CloudFormation

1. Nel riquadro di navigazione dei servizi della console AWS OpsWorks Stacks scegli App.
2. Nella pagina Apps (App), selezionare Node.js Demo App (Applicazione demo Node.js), quindi in Actions (Operazioni), scegliere delete (eliminazione). Quando viene richiesto di confermare l'operazione, scegli Delete (Elimina). AWS OpsWorks Stacks eliminerà l'app.

Per eliminare lo stack

Poiché hai creato lo stack eseguendo un AWS CloudFormation modello, puoi eliminare lo stack, inclusi il layer, l'istanza, il profilo di istanza e il gruppo di sicurezza creato dal modello, nella console. AWS CloudFormation

1. Aprire la AWS CloudFormation console.
2. Nella dashboard della AWS CloudFormation console, seleziona lo stack che hai creato. Nel menu Actions (Operazioni), scegliere Delete Stack (Elimina stack). Quando viene richiesto di confermare, selezionare Yes, Delete (Sì, elimina).
3. Attendere che venga visualizzato DELETE_COMPLETE (Eliminazione completata) nella colonna Status (Stato) relativa allo stack.

Per eliminare la pipeline

1. Aprire la CodePipeline console.

2. Nella CodePipeline dashboard, scegli la pipeline che hai creato per questa procedura dettagliata.
3. Nella pagina della pipeline, scegliere Edit (Modifica).
4. Nella pagina Edit (Modifica), scegliere Delete (Elimina). Quando viene richiesto di confermare, selezionare Delete (Elimina).

AWS CodePipeline con AWS OpsWorks Stacks - Chef 11 Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS CodePipelineti consente di creare pipeline di distribuzione continua che tengono traccia delle modifiche al codice da fonti come CodeCommit Amazon Simple Storage Service (Amazon S3) o. GitHub L'esempio in questo argomento descrive come creare e utilizzare una pipeline semplice CodePipeline come strumento di distribuzione per il codice da eseguire su AWS OpsWorks livelli Stacks. In questo esempio, create una pipeline per una semplice app PHP, quindi chiedete a AWS OpsWorks Stacks di eseguire l'app su tutte le istanze di un livello in uno stack di Chef 11.10 (in questo caso, una singola istanza).

> **Note**
>
> In questo argomento viene descritto come utilizzare una pipeline per eseguire e aggiornare un'app su uno stack Chef 11.10. Per ulteriori informazioni su come utilizzare una pipeline per eseguire e aggiornare un'app su uno stack Chef 12, consulta AWS CodePipeline con AWS OpsWorks pile - Chef 12 Stacks. I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi Come svuotare un bucket S3? o Come eliminare un bucket S3?.

Argomenti

- Prerequisiti

- Altri scenari supportati
- Fase 1: Creazione di stack, livello e istanza in AWS OpsWorks Stacks
- Passaggio 2: Caricamento del codice dell'app in un bucket Amazon S3
- Passaggio 3: aggiungere l'app a AWS OpsWorks Stacks
- Passaggio 4: creazione di una pipeline in CodePipeline
- Fase 5: Verifica della distribuzione dell'app in Stacks AWS OpsWorks
- Fase 6 (opzionale): Aggiornamento del codice dell'app per impostare CodePipeline per la ridistribuzione automatica dell'app
- Fase 7 (opzionale): Pulizia delle risorse

Prerequisiti

Prima di iniziare questa procedura guidata, verifica di disporre delle autorizzazioni di amministratore per eseguire tutte le attività seguenti. Potete far parte di un gruppo a cui è applicata la AdministratorAccesspolicy oppure potete far parte di un gruppo che dispone delle autorizzazioni e delle politiche mostrate nella tabella seguente. Come procedura consigliata in materia di sicurezza, è consigliabile appartenere a un gruppo che dispone delle autorizzazioni necessarie per eseguire le seguenti attività, anziché assegnare le autorizzazioni necessarie ai singoli utenti.

Per ulteriori informazioni sulla creazione di un gruppo di sicurezza in IAM e sull'assegnazione delle autorizzazioni al gruppo, consulta Creazione di gruppi di utenti IAM. Per ulteriori informazioni sulla gestione delle autorizzazioni AWS OpsWorks Stacks, consulta Best Practices: Managing Permissions.

Autorizzazioni	Policy consigliata da collegare a un gruppo
Crea e modifica pile, livelli e istanze in Stacks. AWS OpsWorks	AWSOpsWorks_FullAccess
Creare, modificare ed eseguire modelli in AWS CloudFormation.	AmazonCloudFormationFullAccess
Crea, modifica e accedi ai bucket Amazon S3.	Amazon S3 FullAccess

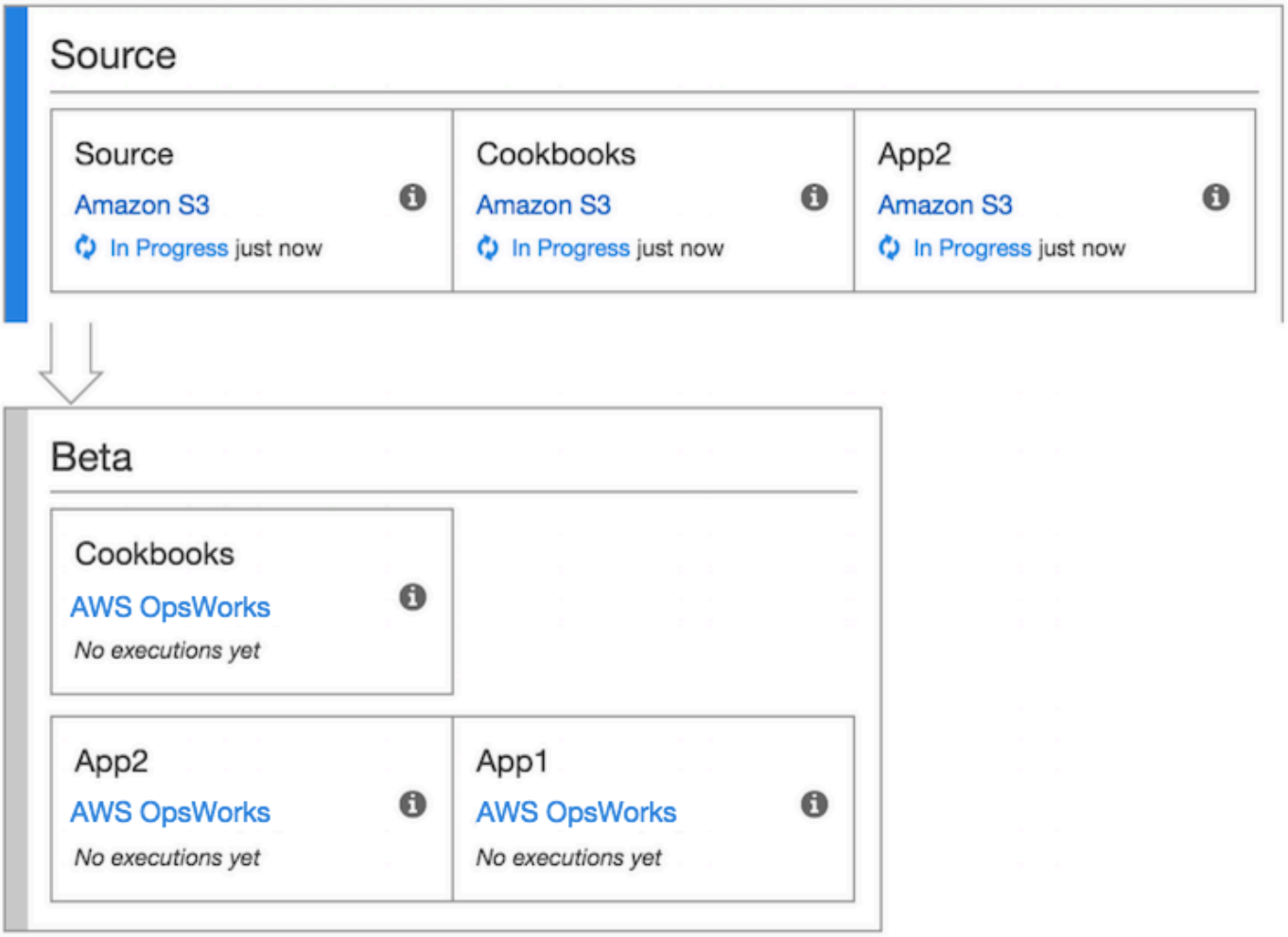
Autorizzazioni	Policy consigliata da collegare a un gruppo
Crea, modifica ed esegui pipeline CodePipeline, in particolare pipeline che utilizzano AWS OpsWorks Stacks come provider.	AWSCodePipeline_FullAccess

È inoltre necessario disporre di una coppia di EC2 chiavi Amazon. Ti verrà richiesto di fornire il nome di questa key pair quando esegui il AWS CloudFormation modello che crea lo stack, il layer e l'istanza di esempio in questa procedura dettagliata. Per ulteriori informazioni su come ottenere una coppia di chiavi nella EC2 console Amazon, consulta Create a Key Pair nella EC2 documentazione di Amazon. La key pair deve trovarsi nella regione Stati Uniti orientali (Virginia settentrionale). Puoi anche utilizzare una coppia di chiavi esistente se ne hai già una in tale regione.

Altri scenari supportati

Questa procedura guidata crea una pipeline semplice che include una fase Source (Origine) e una fase Deploy (Distribuzione). Tuttavia, puoi creare pipeline più complesse che utilizzano AWS OpsWorks Stacks come provider. Di seguito sono illustrati alcuni esempi di pipeline e scenari supportati:

- È possibile modificare una pipeline per aggiungere un libro di ricette Chef alla fase Source (Origine) e una destinazione associata per i libri di ricette aggiornati alla fase Deploy (Distribuzione). In questo caso, si aggiunge un'operazione Deploy (Distribuzione) che attiva l'aggiornamento dei libri di ricette quando vengono apportate modifiche all'origine. Il libro di ricette aggiornato viene distribuito prima dell'app.
- Puoi creare una pipeline complessa, con libri di cucina personalizzati e più app, e distribuirla in uno stack Stacks. AWS OpsWorks La pipeline tiene traccia delle modifiche alle origini sia dell'applicazione che del libro di ricette ed esegue di nuovo la distribuzione in caso di modifiche. Di seguito è illustrato un esempio di una pipeline complessa simile:



Per ulteriori informazioni su come lavorare con CodePipeline, consulta la documentazione. CodePipeline

Fase 1: Creazione di stack, livello e istanza in AWS OpsWorks Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per utilizzare AWS OpsWorks Stacks come provider di distribuzione per una pipeline, devi prima avere uno stack, un layer e almeno un'istanza nel layer. Sebbene sia possibile creare uno stack in AWS OpsWorks Stacks seguendo le istruzioni in Getting Started with Linux Stacks o Getting Started with Windows Stacks, per risparmiare tempo, questo esempio utilizza un AWS CloudFormation modello per creare uno stack, un layer e un'istanza di Chef 11.10 basati su Linux. L'istanza creata in base a questo modello esegue Amazon Linux 2016.03 ed è di tipo `c3.large`.

> **Important**
>
> Il AWS CloudFormation modello deve essere archiviato ed eseguito nella stessa regione del bucket Amazon S3 in cui successivamente carichi l'app e nella stessa regione in cui successivamente crei la pipeline. CodePipeline Al momento, il provider AWS OpsWorks Stacks è CodePipeline supportato solo nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1). Tutte le risorse in questa procedura dettagliata devono essere create nella regione Stati Uniti orientali (Virginia settentrionale).
>
> Se si verifica un errore di creazione dello stack, la causa potrebbe essere il raggiungimento del numero massimo permesso di ruoli IAM per l'account. Può verificarsi un errore nella creazione dello stack anche se l'account non è in grado di avviare istanze di tipo `c3.large`. Ad esempio, se utilizzi il piano AWS gratuito, potresti ricevere un errore del tipo. `Root device type: must be included in EBS` Se il tuo account presenta limitazioni sui tipi di istanze che puoi creare, come le limitazioni imposte dal piano AWS gratuito, prova a modificare il valore del `InstanceType` parametro nel blocco di istanze del modello con un tipo di istanza utilizzabile dal tuo account.

Per creare uno stack, un layer e un'istanza utilizzando AWS CloudFormation

1. Copia il seguente AWS CloudFormation modello in un nuovo documento di testo semplice. Salvate il file in una posizione comoda sul computer locale e denominatelo NewOpsWorksStack.template o con un altro nome adatto alle vostre esigenze.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Mappings": {
    "Region2Principal": {
      "us-east-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
```

```
      "us-west-2": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "us-west-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "eu-west-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "ap-southeast-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "ap-northeast-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "ap-northeast-2": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "ap-southeast-2": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "sa-east-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      },
      "cn-north-1": {
        "EC2Principal": "ec2.amazonaws.com.cn",
        "OpsWorksPrincipal": "opsworks.amazonaws.com.cn"
      },
      "eu-central-1": {
        "EC2Principal": "ec2.amazonaws.com",
        "OpsWorksPrincipal": "opsworks.amazonaws.com"
      }
    }
  },
  "Parameters": {
    "EC2KeyPairName": {
```

```
  "Type": "String",
  "Description": "The name of an existing EC2 key pair that allows you to use SSH
to connect to the OpsWorks instance."
 }
 },
 "Resources": {
"CPOpsDeploySecGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription" : "Lets you manage OpsWorks instances deployed to by
CodePipeline"
  }
},
"CPOpsDeploySecGroupIngressHTTP": {
  "Type": "AWS::EC2::SecurityGroupIngress",
  "Properties" : {
    "IpProtocol" : "tcp",
       "FromPort" : "80",
       "ToPort" : "80",
       "CidrIp" : "0.0.0.0/0",
 "GroupId": {
   "Fn::GetAtt": [
     "CPOpsDeploySecGroup", "GroupId"
   ]
 }
     }
},
"CPOpsDeploySecGroupIngressSSH": {
  "Type": "AWS::EC2::SecurityGroupIngress",
  "Properties" : {
    "IpProtocol" : "tcp",
       "FromPort" : "22",
       "ToPort" : "22",
       "CidrIp" : "0.0.0.0/0",
 "GroupId": {
   "Fn::GetAtt": [
     "CPOpsDeploySecGroup", "GroupId"
   ]
 }
  }
},
"MyStack": {
     "Type": "AWS::OpsWorks::Stack",
     "Properties": {
```

```
        "Name": {
          "Ref": "AWS::StackName"
        },
        "ServiceRoleArn": {
          "Fn::GetAtt": [
            "OpsWorksServiceRole",
            "Arn"
          ]
        },
  "ConfigurationManager" : { "Name": "Chef","Version": "11.10" },
  "DefaultOs": "Amazon Linux 2016.03",
        "DefaultInstanceProfileArn": {
          "Fn::GetAtt": [
            "OpsWorksInstanceProfile",
            "Arn"
          ]
        }
      }
    },
    "MyLayer": {
      "Type": "AWS::OpsWorks::Layer",
      "Properties": {
        "StackId": {
          "Ref": "MyStack"
        },
        "Name": "MyLayer",
        "Type": "php-app",
  "Shortname": "mylayer",
        "EnableAutoHealing": "true",
        "AutoAssignElasticIps": "false",
        "AutoAssignPublicIps": "true",
  "CustomSecurityGroupIds": [
    {
      "Fn::GetAtt": [
              "CPOpsDeploySecGroup", "GroupId"
      ]
          }
        ]
      },
      "DependsOn": [
        "MyStack",
        "CPOpsDeploySecGroup"
      ]
    },
```

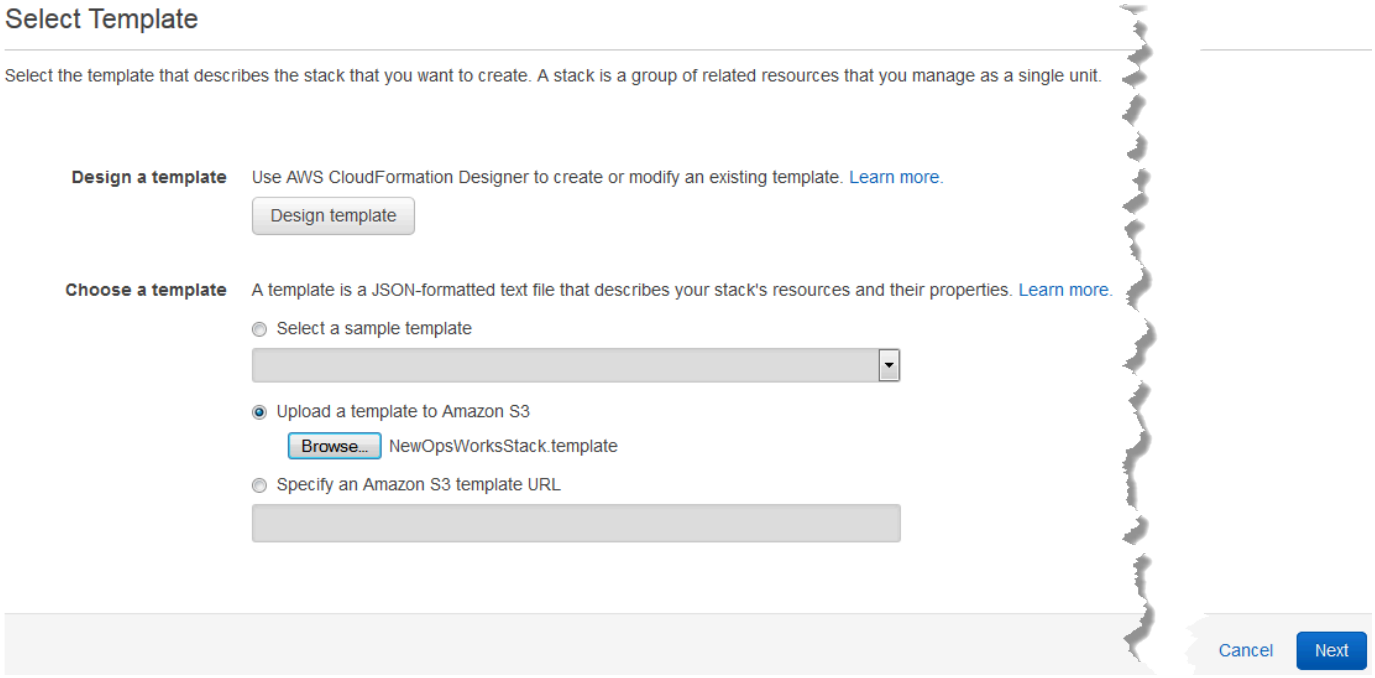
```
    "OpsWorksServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  {
                    "Fn::FindInMap": [
                      "Region2Principal",
                      {
                        "Ref": "AWS::Region"
                      },
                      "OpsWorksPrincipal"
                    ]
                  }
                ]
              },
              "Action": [
                "sts:AssumeRole"
              ]
            }
          ]
        },
        "Path": "/",
        "Policies": [
          {
            "PolicyName": "opsworks-service",
            "PolicyDocument": {
              "Statement": [
                {
                  "Effect": "Allow",
                  "Action": [
                    "ec2:*",
                    "iam:PassRole",
                    "cloudwatch:GetMetricStatistics",
                    "elasticloadbalancing:*"
                  ],
                  "Resource": "*"
                }
              ]
            }
```

```
          }
        ]
      }
    },
    "OpsWorksInstanceProfile": {
      "Type": "AWS::IAM::InstanceProfile",
      "Properties": {
        "Path": "/",
        "Roles": [
          {
            "Ref": "OpsWorksInstanceRole"
          }
        ]
      }
    },
    "OpsWorksInstanceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  {
                    "Fn::FindInMap": [
                      "Region2Principal",
                      {
                        "Ref": "AWS::Region"
                      },
                      "EC2Principal"
                    ]
                  }
                ]
              },
              "Action": [
                "sts:AssumeRole"
              ]
            }
          ]
        },
        "Path": "/",
  "Policies": [
          {
```

```
            "PolicyName": "s3-get",
            "PolicyDocument": {
              "Version": "2012-10-17",
              "Statement": [
                {
                  "Effect": "Allow",
                  "Action": [
                    "s3:GetObject"
                  ],
                  "Resource": "*"
                }
              ]
            }
          }
        ]
      }
    },
    "myinstance": {
      "Type": "AWS::OpsWorks::Instance",
      "Properties": {
        "LayerIds": [
          {
            "Ref": "MyLayer"
          }
        ],
        "StackId": {
          "Ref": "MyStack"
        },
        "InstanceType": "c3.large",
        "SshKeyName": {
    "Ref": "EC2KeyPairName"
  }
      }
    }
  },
  "Outputs": {
    "StackId": {
      "Description": "Stack ID for the newly created AWS OpsWorks stack",
      "Value": {
        "Ref": "MyStack"
      }
    }
  }
```

```
}
```

2. Accedi a AWS Management Console e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformazione.
3. Nella AWS CloudFormation home page, scegli Crea stack.
4. Nella pagina Select Template (Seleziona modello) nell'area Choose a template (Scegliere un modello) scegliere Upload a template to Amazon S3 (Carica un modello in Amazon S3) e quindi Browse (Sfoglia).
5. Vai al AWS CloudFormation modello che hai salvato nel passaggio 1, quindi scegli Apri. Nella pagina Select Template (Seleziona modello), selezionare Next (Avanti).

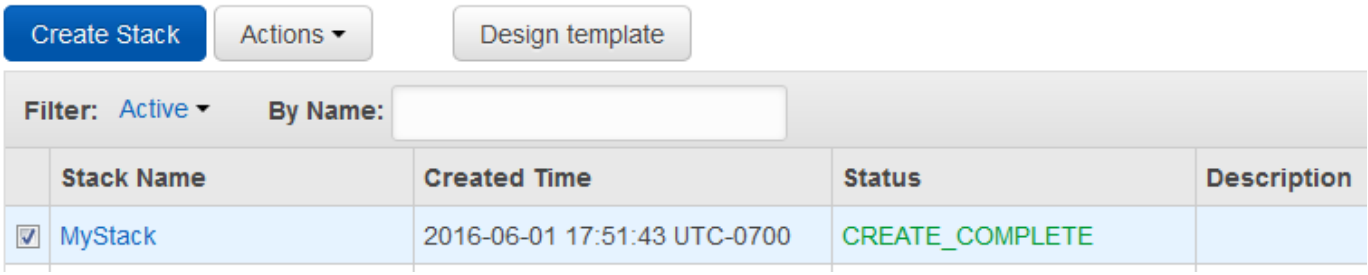


6. Nella pagina Specificare i dettagli, assegna un nome allo stack MyStacko a qualsiasi nome dello stack univoco per il tuo account. Se si sceglie un nome diverso per lo stack, modificare il nome dello stack in tutta questa procedura guidata.
7. Nell'area Parametri, fornisci il nome di una coppia di EC2 key pair che desideri utilizzare per accedere all'istanza AWS OpsWorks Stacks dopo che è stata creata. Scegli Next (Successivo).
8. Nella pagina Opzioni, scegli Avanti. Le impostazioni in questa pagina non sono necessarie per questa procedura guidata.
9. Il AWS CloudFormation modello utilizzato in questa procedura dettagliata crea ruoli IAM, un profilo di istanza e un'istanza.

> **Important**
>
> Prima di scegliere Create, scegli Costo per stimare gli addebiti che potresti dover sostenere AWS per la creazione di risorse con questo modello.

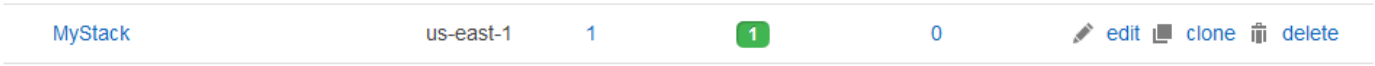
Se la creazione di risorse IAM è accettabile, seleziona la casella di controllo Riconosco che questo modello potrebbe indurre AWS CloudFormation a creare risorse IAM, quindi scegli Crea. Se la creazione di risorse IAM non è accettabile, non puoi continuare con questa procedura.

10. Nel AWS CloudFormation pannello di controllo è possibile visualizzare l'avanzamento della creazione dello stack. Prima di continuare con la fase successiva, attendere che venga visualizzato il messaggio CREATE_COMPLETE (CREAZIONE_COMPLETATA) nella colonna Status (Stato).



Per verificare la creazione di stack in Stacks AWS OpsWorks

1. Apri la AWS OpsWorks console all'indirizzo. https://console.aws.amazon.com/opsworks/
2. Nella dashboard AWS OpsWorks Stacks, visualizza lo stack che hai creato.



3. Aprire lo stack e visualizzare il livello e l'istanza. Osserva che il layer e l'istanza sono stati creati con i nomi e gli altri metadati forniti nel AWS CloudFormation modello. Sei pronto per caricare la tua app in un bucket Amazon S3.

Passaggio 2: Caricamento del codice dell'app in un bucket Amazon S3

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Poiché è necessario fornire un collegamento al repository di codice come parte della configurazione della pipeline, prepara il repository di codice prima di creare la pipeline. Questa procedura dettagliata prevede il caricamento di un'app PHP in un bucket Amazon S3.

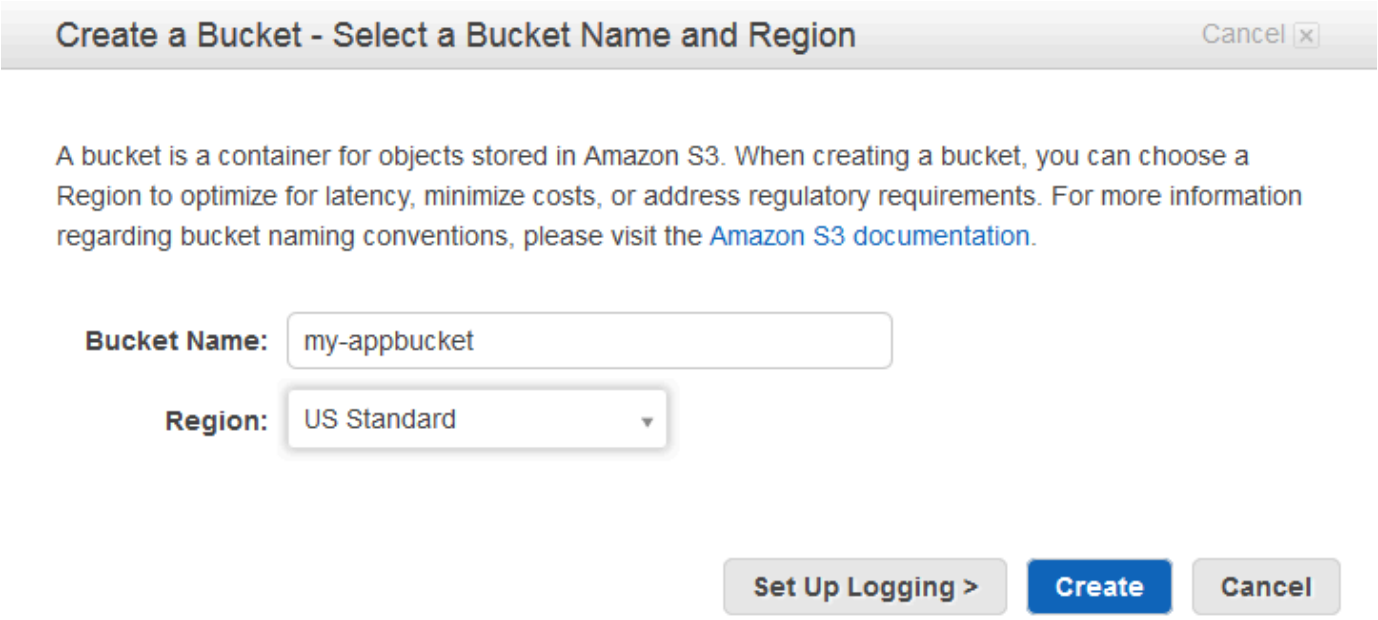
Sebbene sia CodePipeline possibile utilizzare il codice direttamente da GitHub o CodeCommit come sorgente, questa procedura dettagliata dimostra come utilizzare un bucket Amazon S3. Il bucket Amazon S3 consente di rilevare CodePipeline le modifiche al codice dell'app e di distribuire automaticamente l'app modificata. Se desideri, puoi utilizzare un bucket esistente. Assicurati che il bucket soddisfi i criteri descritti in Simple CodePipeline Pipeline Walkthrough (Amazon S3 Bucket) nella documentazione. CodePipeline

> **Important**
>
> Il bucket Amazon S3 deve trovarsi nella stessa regione in cui crei la pipeline in un secondo momento. Al momento, il provider AWS OpsWorks Stacks è CodePipeline supportato solo nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1). Tutte le risorse in questa procedura dettagliata devono essere create nella regione Stati Uniti orientali (Virginia settentrionale). È inoltre necessario che il bucket abbia una versione del codice sorgente, poiché CodePipeline richiede una versione del codice sorgente. Per ulteriori informazioni, consultare Uso della funzione Versioni multiple.

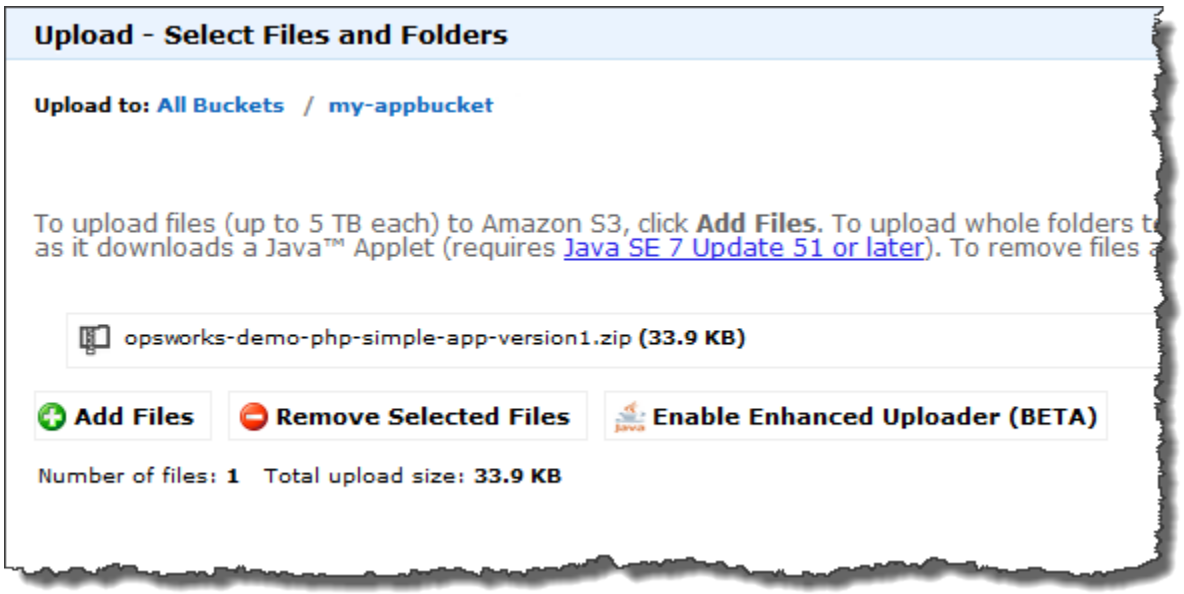
Per caricare l'app in un bucket Amazon S3

1. Dal GitHub sito Web, scarica un file ZIP dell'app PHP di esempio AWS OpsWorks Stacks e salvalo in una posizione comoda sul tuo computer locale.

2. Assicurarsi che `index.php` e la cartella `ASSETS` si trovino al livello root del file ZIP scaricato. In caso contrario, decomprimere il file e creare un nuovo file ZIP con questi file al livello root.
3. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
4. Scegli Crea bucket.
5. Nella pagina Create a Bucket - Select a Bucket Name and Region (Creare un bucket - Selezionare un nome di bucket e una regione), in Bucket Name (Nome bucket), digitare il nome univoco del bucket. I nomi di bucket devono essere univoci per tutti AWS gli account, non solo per il dominio. Questa procedura guidata utilizza il nome **`my-appbucket`**, ma è possibile utilizzare `my-appbucket-`*`yearmonthday`* per rendere univoco il nome del bucket. Nell'elenco a discesa Region (Regione), scegliere US Standard (Stati Uniti standard), quindi scegliere Create (Crea). US Standard (Stati Uniti standard) equivale a `us-east-1`.



6. Scegliere il bucket creato nell'elenco All Buckets (Tutti i bucket).
7. Nella pagina del bucket scegliere Upload (Carica).
8. Nella pagina Upload - Select Files and Folders (Carica - Seleziona file e cartelle), scegliere Add Files (Aggiungi file). Cercare il file ZIP salvato nella fase 1, scegliere Open (Apri), quindi scegliere Start Upload (Avvia caricamento).



9. Al termine del caricamento, selezionare il file ZIP nell'elenco di file nel bucket, quindi scegliere Properties (Proprietà).
10. Nel riquadro Properties (Proprietà), copiare il collegamento al file ZIP, quindi prendere nota del collegamento. La parte di collegamento con il nome del bucket e il nome del file ZIP sarà necessaria per creare la pipeline.

Passaggio 3: aggiungere l'app a AWS OpsWorks Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Prima di creare una pipeline in CodePipeline, aggiungi l'app di test PHP a Stacks. AWS OpsWorks Quando crei la pipeline, dovrai selezionare l'app che hai aggiunto a Stacks. AWS OpsWorks

Tieni pronto il link al bucket Amazon S3 del passaggio 10 della procedura precedente. Per completare questa procedura, è necessario il collegamento al bucket in cui hai archiviato l'app di test.

Per aggiungere un'app a Stacks AWS OpsWorks

1. Nella console AWS OpsWorks Stacks MyStack, apri e nel pannello di navigazione scegli App.
2. Scegliere Add app (Aggiungi app).
3. Nella pagina Add app (Aggiungi app), fornire le seguenti informazioni:

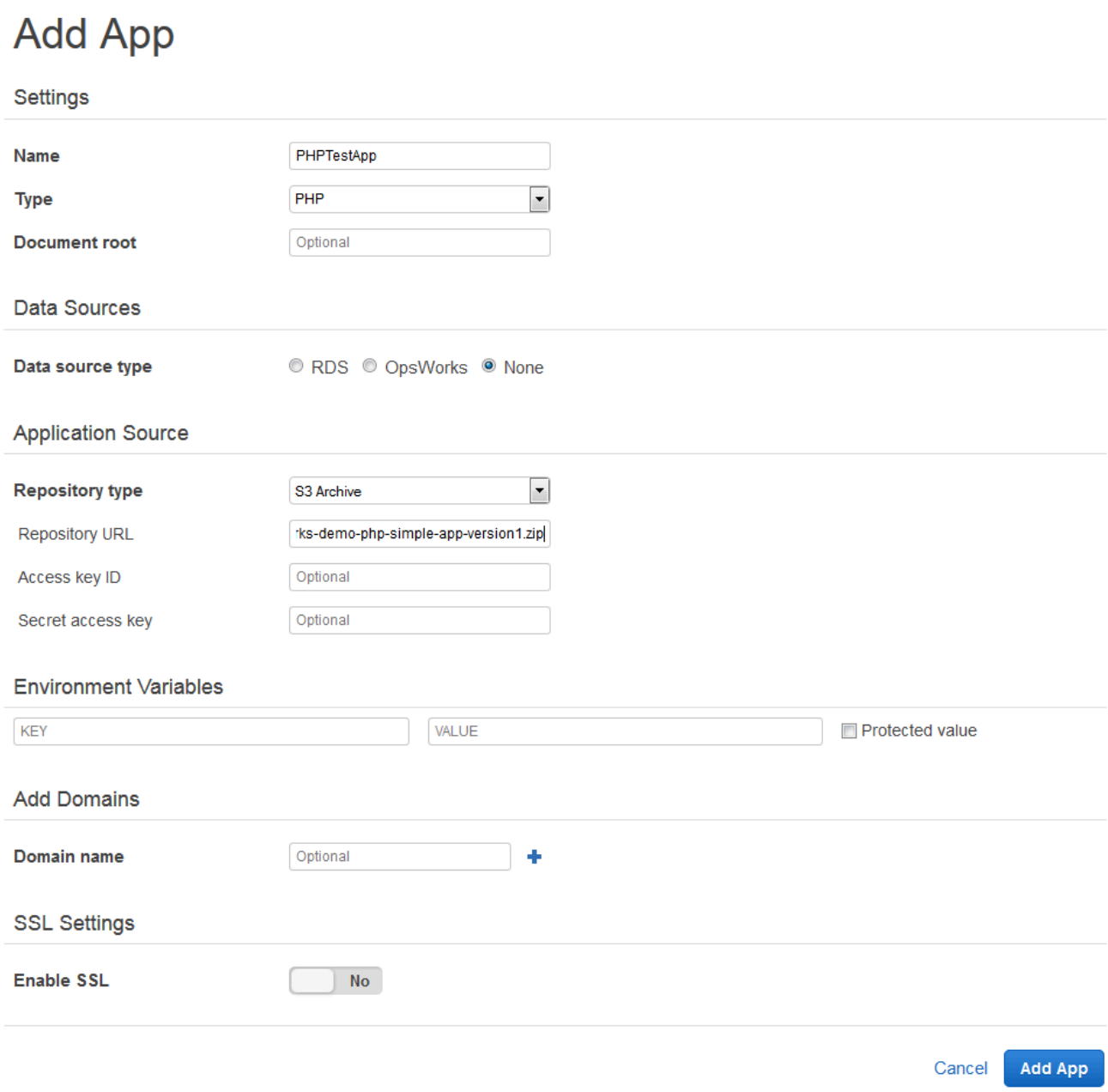
   a. Specificare un nome per l'app. Questa procedura guidata utilizza il nome `PHPTestApp`.

   b. Nell'elenco a discesa Type (Tipo) scegliere PHP.

   c. In Data source type (Tipo di origine dati), scegliere None (Nessuno). Questa app non richiede un'origine dati o un database esterno.

   d. Nell'elenco a discesa Repository type (Tipo di repository), scegliere S3 Archive (Archivio S3).

   e. Nella casella di testo Repository URL (URL repository), incollare l'URL copiato nella fase 10 della Passaggio 2: Caricamento del codice dell'app in un bucket Amazon S3. Il modulo dovrebbe essere simile al seguente:



4. Non è necessario modificare altre impostazioni in questo modulo. Scegliere Add App (Aggiungi app).
5. Quando l'PHPTestapp App appare nell'elenco della pagina App, continua con la procedura successiva,[Passaggio 4: creazione di una pipeline in CodePipeline](#).

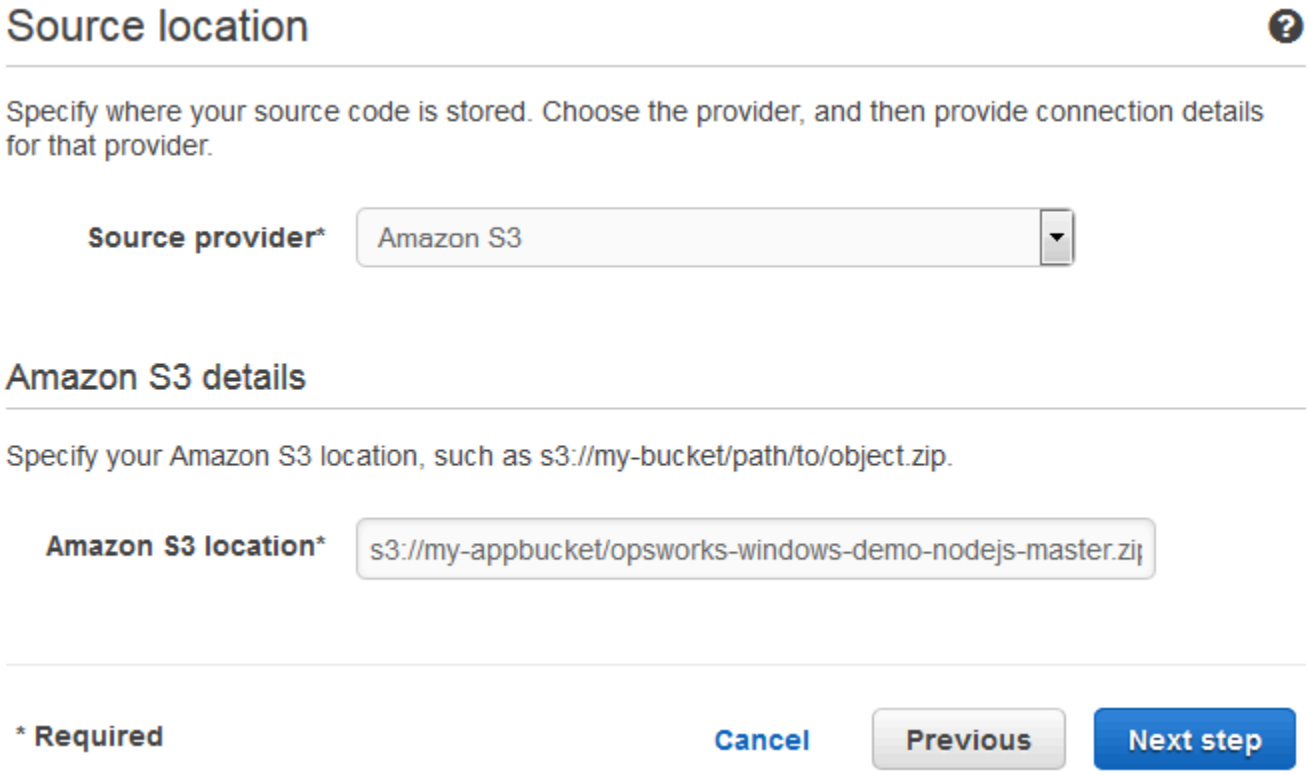
Passaggio 4: creazione di una pipeline in CodePipeline

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

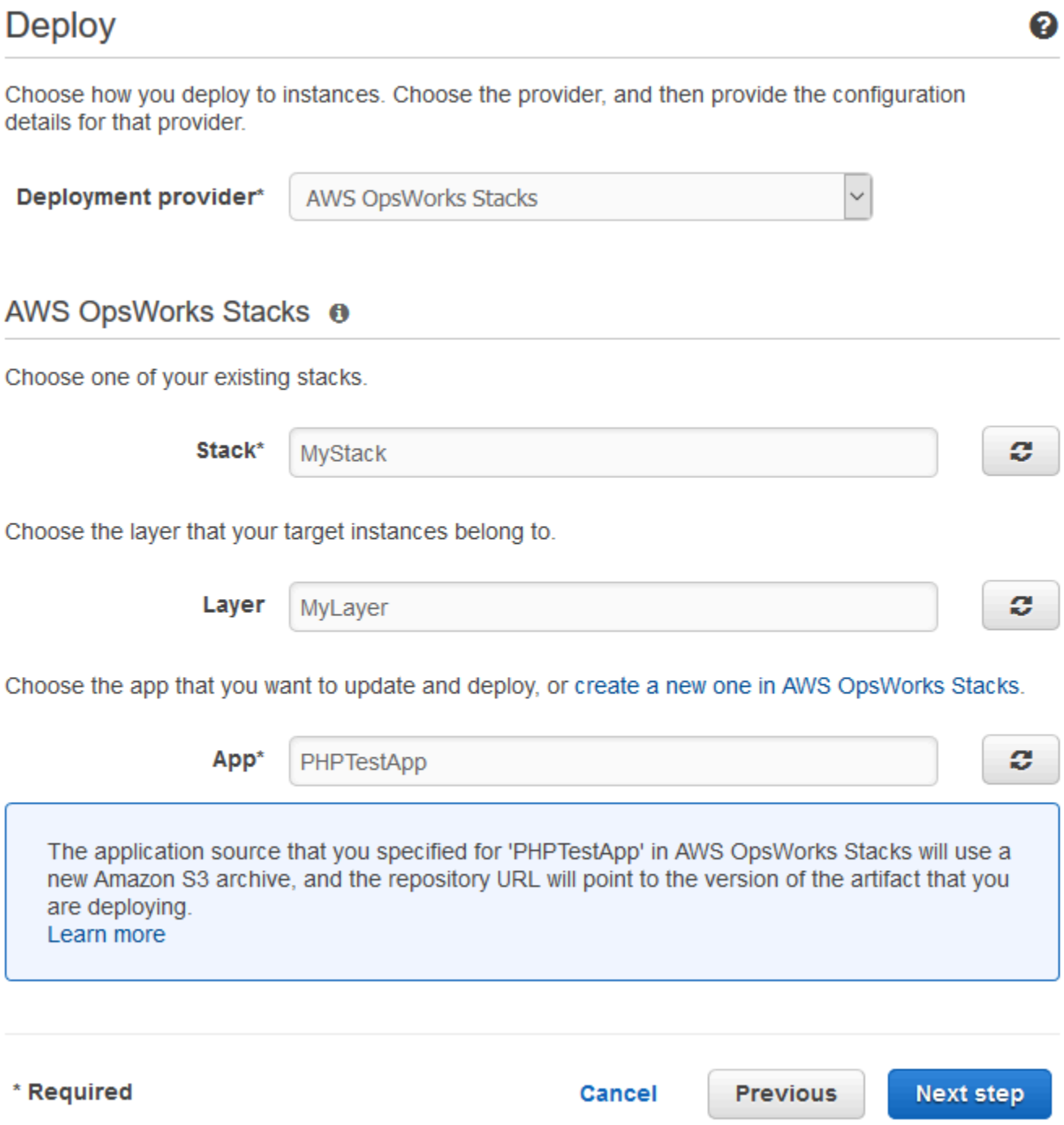
Dopo aver configurato uno stack con un layer e almeno un'istanza configurata in AWS OpsWorks Stacks, crea una pipeline utilizzando AWS OpsWorks Stacks come provider per distribuire app o libri di cucina Chef nelle CodePipeline tue risorse Stacks. AWS OpsWorks

Per creare una pipeline

1. Apri la console all'indirizzo. CodePipeline https://console.aws.amazon.com/codepipeline/
2. Scegliere Create pipeline (Crea pipeline).
3. Nella pagina Getting started with CodePipeline (Nozioni di base su CodePipeline), digitare **`MyOpsWorksPipeline`** o qualsiasi altro nome di pipeline purché sia univoco per l'account in uso, quindi scegliere Next step (Fase successiva).
4. Nella pagina Source Location (Posizione di origine), selezionare Amazon S3 nell'elenco a discesa Source provider (Provider origine).
5. Nell'area dei dettagli di Amazon S3, digita il percorso del bucket Amazon S3, nel formato. **`s3://bucket-name/file name`** Fare riferimento al collegamento annotato nella fase 10 della Passaggio 2: Caricamento del codice dell'app in un bucket Amazon S3. In questa procedura guidata, il percorso è `s3://my-appbucket/opsworks-demo-php-simple-app-version1.zip`. Seleziona Fase successiva.



6. Nella pagina Build (Crea), scegliere No Build (Nessuna creazione) nell'elenco a discesa, quindi scegliere Next step (Fase successiva).
7. Nella pagina Deploy (Distribuzione), scegliere AWS OpsWorks Stacks come provider di distribuzione.



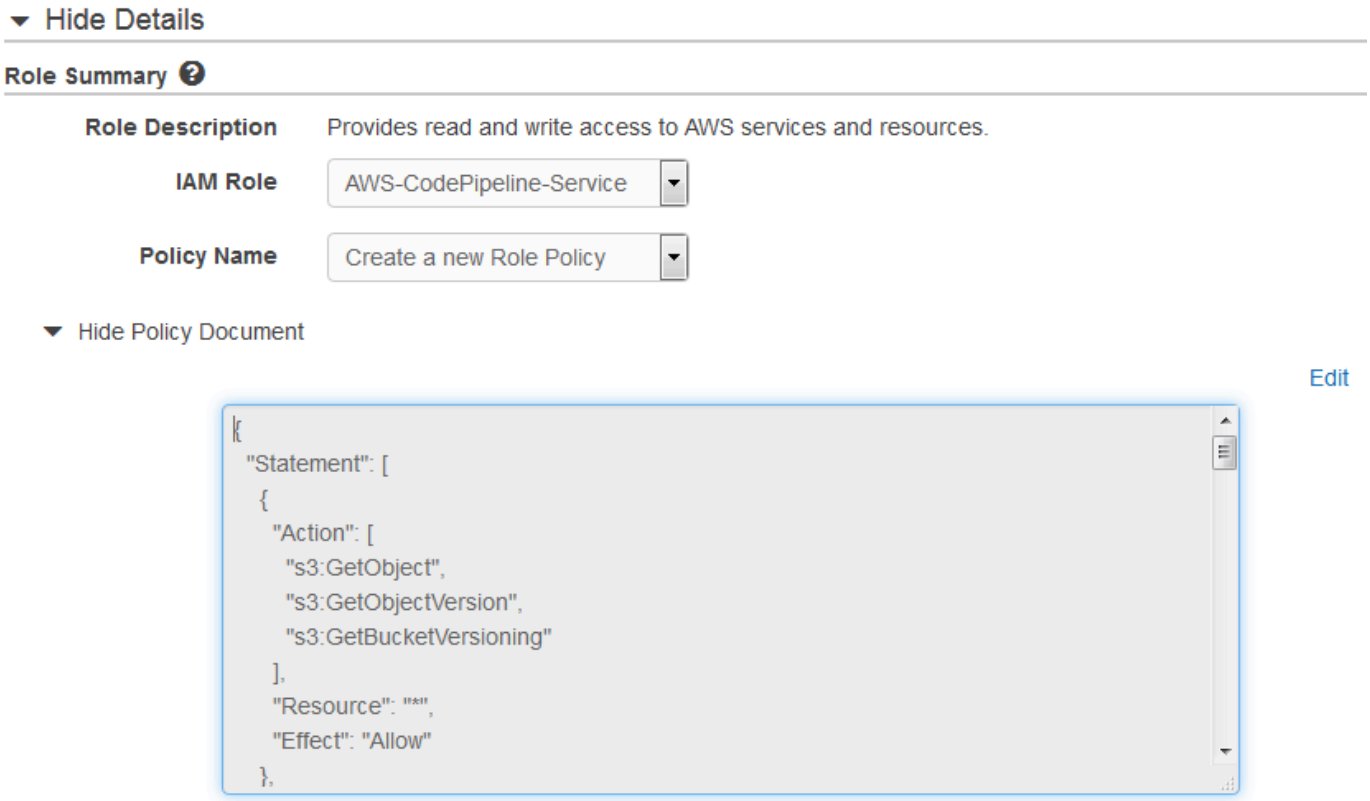
8. Nel campo Stack, digitare `MyStack` oppure il nome dello stack creato nella Fase 1: Creazione di stack, livello e istanza in AWS OpsWorks Stacks.
9. Nel campo Layer (Livello), digitare `MyLayer` oppure il nome del livello creato nella Fase 1: Creazione di stack, livello e istanza in AWS OpsWorks Stacks.

10. Nel campo App, seleziona l'app che hai caricato su Amazon S3Passaggio 2: Caricamento del codice dell'app in un bucket Amazon S3, quindi scegli Passaggio successivo.
11. Nella pagina AWS Service Role (Ruolo servizio AWS), scegliere Create Role (Crea ruolo).

    Si apre una nuova finestra con una pagina della console IAM che descrive il ruolo che verrà creato per te,`AWS-CodePipeline-Service`. Nell'elenco a discesa Policy name (Nome policy), scegliere Create new policy (Crea una nuova policy). Assicurarsi che il documento di policy abbia il contenuto seguente. Scegliere Edit (Modifica) per modificare il documento delle policy, se necessario.

```
{
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetBucketVersioning"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "opsworks:*",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

    Dopo aver modificato il documento di policy, scegliere Allow (Consenti). Le modifiche verranno visualizzate nella console IAM.



> **Note**
>
> Se la creazione del ruolo fallisce, potrebbe essere perché disponi già di un ruolo IAM denominato AWS- CodePipeline -Service. Se hai utilizzato il ruolo AWS- CodePipeline -Service prima di maggio 2016, il ruolo potrebbe non disporre delle autorizzazioni per utilizzare AWS OpsWorks Stacks come fornitore di distribuzione; in questo caso, devi aggiornare l'informativa sulla politica come mostrato in questo passaggio. Se viene visualizzato un messaggio di errore, tornare all'inizio di questa fase e scegliere Use existing role (Utilizza un ruolo esistente) anziché Create role (Crea ruolo). Se si utilizza un ruolo esistente, il ruolo deve avere una policy collegata che include le autorizzazioni indicate in questa fase. Per ulteriori informazioni sul ruolo del servizio e sulla relativa istruzione della policy, consulta l'argomento relativo alla modifica di una policy per un ruolo del servizio IAM.

12. Se il processo di creazione del ruolo ha esito positivo, la pagina IAM si chiuderà e tornerai alla pagina AWS Service Role. Seleziona Fase successiva.
13. Nella pagina Review your pipeline (Verifica pipeline), verificare le selezioni effettuate nella pagina, quindi scegliere Create pipeline (Crea pipeline).

We will create your pipeline with the following resources.

Source Stage

Source provider Amazon S3

Amazon S3 location s3://my-appbucket0/opsworks-demo-php-simple-app-version1.zip

Build Stage

Build provider No Build

Beta Stage

Deployment provider AWS OpsWorks

Stack MyStack

App PHPTestApp

Layer MyLayer

Pipeline settings

Pipeline name MyOpsWorksPipeline

Artifact location s3://codepipeline-us-east-
AWS CodePipeline will use this existing S3 bucket to store artifacts for this pipeline. Depending on the size of your artifacts, you might be charged for storage costs. For more information, see Amazon S3 storage pricing.

Role name AWS-CodePipeline-Service

To save this configuration with these resources, choose Create pipeline.

Would you like to create this pipeline?

Cancel Previous Create pipeline

14. Quando la pipeline è pronta, dovrebbe iniziare a individuare il codice sorgente e a distribuire l'app allo stack automaticamente. Questo processo può richiedere alcuni minuti.

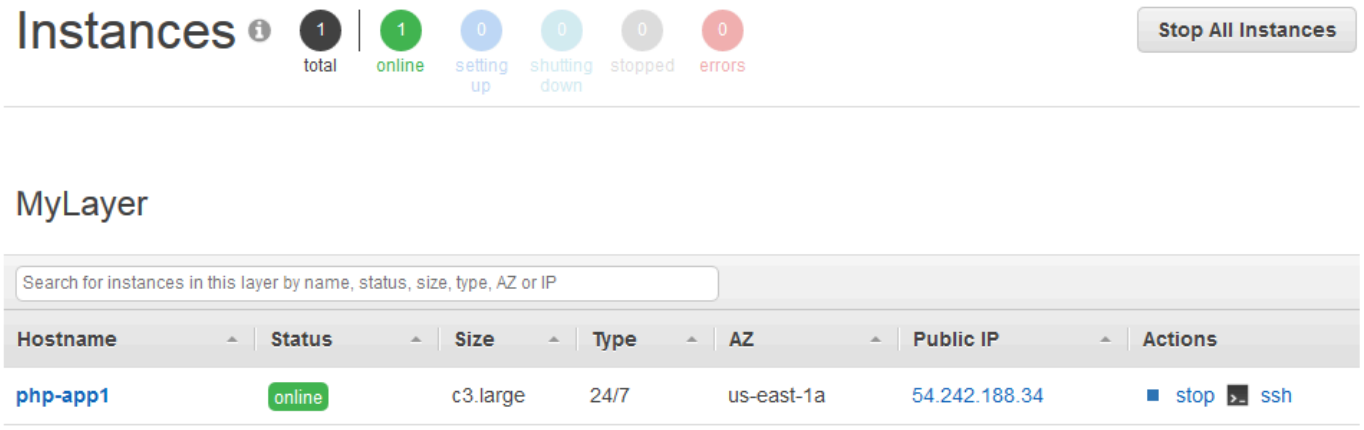
Fase 5: Verifica della distribuzione dell'app in Stacks AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

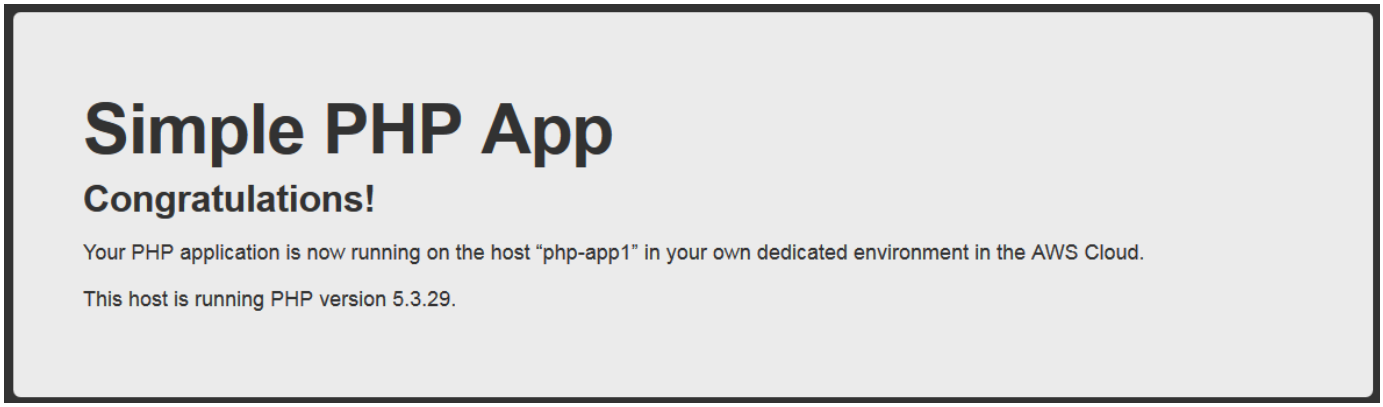
Per verificare di CodePipeline aver distribuito l'app PHP nel tuo stack, accedi all'istanza che hai creato. Fase 1: Creazione di stack, livello e istanza in AWS OpsWorks Stacks Dovresti essere in grado di visualizzare e utilizzare l'app Web PHP.

Per verificare la distribuzione dell'app nella tua istanza Stacks AWS OpsWorks

1. Apri la AWS OpsWorks console all'indirizzo https://console.aws.amazon.com/opsworks/.
2. Nella dashboard AWS OpsWorks Stacks, scegli MyStack, quindi scegli MyLayer.
3. Nel riquadro di navigazione, scegliere Instances (Istanze), quindi scegliere l'indirizzo IP pubblico dell'istanza creata per visualizzare l'app Web.



L'app verrà visualizzata in una nuova scheda del browser.



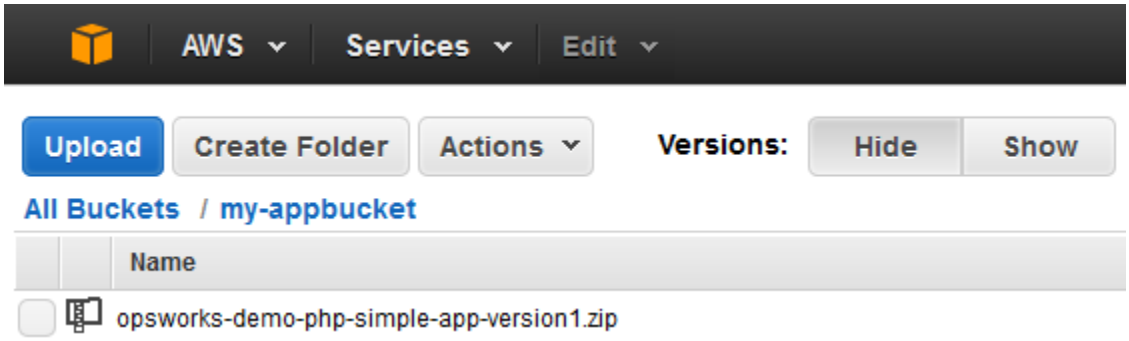
Fase 6 (opzionale): Aggiornamento del codice dell'app per impostare CodePipeline per la ridistribuzione automatica dell'app

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Quando apporti modifiche al codice nelle app o nei libri di cucina che hai distribuito utilizzando CodePipeline, gli artefatti aggiornati verranno distribuiti automaticamente sulle istanze di destinazione (in questo caso, CodePipeline su uno stack di stack di destinazione). AWS OpsWorks Questa sezione illustra la ridistribuzione automatica quando aggiorni il codice nell'app PHP di esempio.

Per modificare il codice dell'app di esempio

1. Accedi a AWS Management Console e apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Aprire il bucket in cui è archiviata l'app PHP di esempio.



3. Selezionare il file ZIP contenente l'app. Nel menu Actions (Operazioni), scegliere Download (Scarica).
4. Nella finestra di dialogo, aprire il menu contestuale facendo clic con il pulsante destro del mouse, scegliere Download (Scarica), quindi salvare il file ZIP in una posizione comoda. Scegli OK.
5. Estrarre il contenuto del file ZIP in un percorso comodo. Potrebbe essere necessario modificare le autorizzazioni per la cartella estratta, le sottocartelle e i contenuti per permettere la modifica. Nella cartella `opsworks-demo-php-simple-app-version1`, aprire il file `index.php` per la modifica.
6. Cercare la frase `Your PHP application is now running`. Sostituire il testo `Your PHP application is now running` con `You've just deployed your first app to AWS OpsWorks with AWS CodePipeline,`. Non modificare le variabili.

```
<body>
    <div class="container">
        <div class="hero-unit">
            <h1>Simple PHP App</h1>
            <h2>Congratulations!</h2>
            <p>You've just deployed your first app to AWS OpsWorks with AWS CodePipeline,</p>
            <p>on the host &ldquo;<?php echo gethostname(); ?>&rdquo; </p>
            <p>in your own dedicated environment in the AWS Cloud.</p>
            <p>This host is running PHP version <?php echo phpversion(); ?>.</p>
        </div>
    </div>

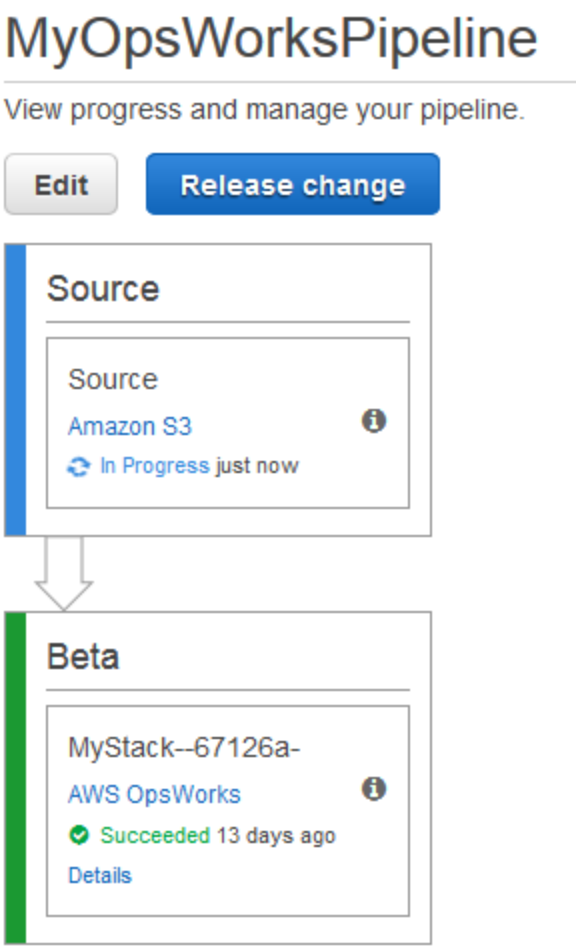
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
    <script src="assets/js/bootstrap.min.js"></script>
</body>
```

7. Salvare e chiudere il file `index.php`.
8. Comprimere la cartella `opsworks-demo-php-simple-app-version1` e salvare il file ZIP in una posizione comoda. Non modificare il nome del file ZIP.
9. Carica il nuovo file ZIP nel tuo bucket Amazon S3. In questa procedura guidata il nome del bucket è `my-appbucket`.

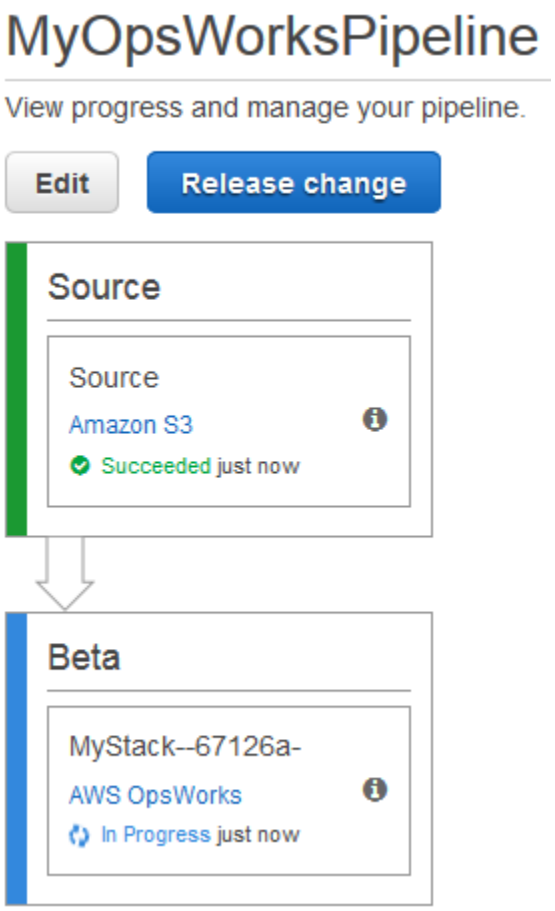
10. Apri la CodePipeline console e apri la pipeline AWS OpsWorks Stacks (). MyOpsWorksPipeline Scegliere Release Change (Rilascia modifica).

    (Puoi attendere che CodePipeline venga rilevata la modifica del codice dalla versione aggiornata dell'app nel tuo bucket Amazon S3. Per farti risparmiare tempo, questa procedura dettagliata ti spiega semplicemente di scegliere Release Change.)

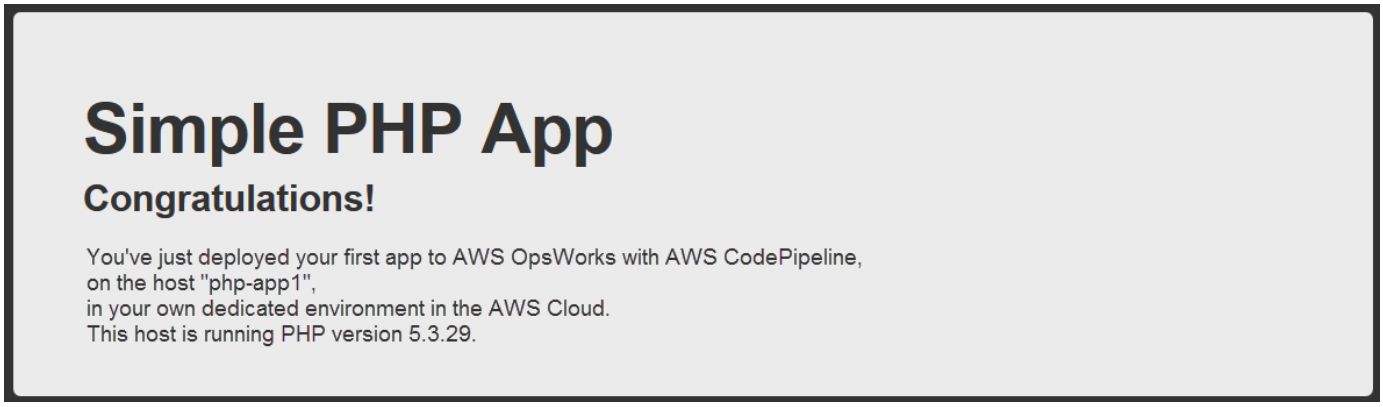
11. Osserva come CodePipeline scorre lungo le fasi della pipeline. Innanzitutto, CodePipeline rileva le modifiche all'artefatto di origine.



    CodePipeline inserisce il codice aggiornato nel tuo stack in Stacks. AWS OpsWorks



12. Quando entrambe le fasi della pipeline sono state completate con successo, apri lo stack in Stacks (). AWS OpsWorks MyStack
13. Nella pagina delle MyStackproprietà, scegli Istanze.
14. Nella colonna Public IP (IP pubblico), scegliere l'indirizzo IP pubblico dell'istanza per visualizzare il testo dell'app aggiornata.



Fase 7 (opzionale): Pulizia delle risorse

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per evitare addebiti indesiderati sul tuo account AWS, puoi eliminare le AWS risorse utilizzate per questa procedura dettagliata. Queste AWS risorse includono lo AWS OpsWorks stack Stacks, il ruolo e il profilo dell'istanza IAM e la pipeline in cui hai creato. CodePipeline Tuttavia, potresti voler continuare a utilizzare queste AWS risorse man mano che continui a saperne di più su AWS OpsWorks Stacks e. CodePipeline Se desideri mantenere queste risorse, hai completato la procedura guidata.

Per eliminare l'app dallo stack

Poiché non hai creato o applicato l'app come parte del tuo AWS CloudFormation modello, elimina l'app di test PHP prima di eliminare lo stack in esso contenuto. AWS CloudFormation

1. Nella console AWS OpsWorks Stacks, nel riquadro di navigazione del servizio scegli App.
2. Nella pagina App, seleziona PHPTestApp, quindi in Azioni scegli elimina. Quando viene richiesto di confermare l'operazione, scegli Delete (Elimina). AWS OpsWorks Stacks eliminerà l'app.

Per eliminare lo stack

Poiché hai creato lo stack eseguendo un AWS CloudFormation modello, puoi eliminare lo stack, inclusi il layer, l'istanza, il profilo di istanza e il gruppo di sicurezza creato dal modello, nella console. AWS CloudFormation

1. Aprire la AWS CloudFormation console.
2. Nella dashboard della AWS CloudFormation console, seleziona lo stack che hai creato (MyStack). Nel menu Actions (Operazioni), scegliere Delete Stack (Elimina stack). Quando viene richiesto di confermare, selezionare Yes, Delete (Sì, elimina).

3. Attendere che venga visualizzato DELETE_COMPLETE (Eliminazione completata) nella colonna Status (Stato) relativa allo stack.

Per eliminare la pipeline

1. Aprire la CodePipeline console.
2. Nella CodePipeline dashboard, scegli la pipeline che hai creato per questa procedura dettagliata.
3. Nella pagina della pipeline, scegliere Edit (Modifica).
4. Nella pagina Edit (Modifica), scegliere Delete (Elimina). Quando viene richiesto di confermare, selezionare Delete (Elimina).

Utilizzo della CLI AWS OpsWorks di Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

L'interfaccia a riga di comando (CLI) di AWS OpsWorks Stacks offre le stesse funzionalità della console e può essere utilizzata per una varietà di attività. La CLI di AWS OpsWorks Stacks fa parte di. AWS CLI Per ulteriori informazioni, tra cui come installare e configurare AWS CLI, consulta What Is the AWS Command Line Interface? . Per una descrizione completa di ogni comando, consulta la guida di riferimento su AWS OpsWorks Stacks.

> **Note**
>
> Se si utilizza una workstation basata su Windows, è inoltre possibile eseguire AWS gli strumenti per Windows per PowerShell eseguire le operazioni AWS OpsWorks Stacks dalla riga di comando. Per ulteriori informazioni, consulta AWS Tools per Windows PowerShell.

AWS OpsWorks I comandi Stacks hanno il seguente formato generale:

```
aws opsworks --region us-west-1 opsworks command-name [--argument1 value] [...]
```

Se il valore di un argomento è un oggetto JSON, devi eseguire l'escape dei caratteri " o il comando potrebbe restituire un errore con il quale viene indicato che il JSON non è valido. Ad esempio, se l'oggetto JSON è `"{"somekey":"somevalue"}"`, devi utilizzare il formato `"{\"somekey\": \"somevalue\"}"`. In alternativa, puoi inserire l'oggetto JSON in un file e utilizzare `file://` per includerlo nella riga di comando. L'esempio seguente crea un'app utilizzando un oggetto source dell'applicazione archiviato in appsource.json.

```
aws opsworks --region us-west-1 create-app --stack-id 8c428b08-a1a1-46ce-a5f8-
feddc43771b8 --name SimpleJSP --type java --app-source file://appsource.json
```

La maggior parte dei comandi restituisce uno o più valori, impacchettati come oggetti JSON. Le seguenti sezioni contengono alcuni esempi. Per una descrizione dettagliata dei valori restituiti per ogni comando, consulta la guida di riferimento di AWS OpsWorks Stacks.

> **Note**
>
> AWS CLI i comandi devono specificare una regione, come mostrato negli esempi. I valori validi per il parametro --region sono illustrati nella seguente tabella. Per semplificare le stringhe di comando AWS OpsWorks Stacks, configura la CLI per specificare la regione predefinita, in modo da poter omettere il parametro. `--region` Se in genere lavori su più endpoint regionali, non configurarli AWS CLI per utilizzare un endpoint regionale predefinito. L'endpoint della regione Canada (Centrale) è disponibile AWS CLI solo nell'API; non è disponibile per gli stack creati in. AWS Management Console Per ulteriori informazioni, consulta la sezione relativa alla configurazione della regione AWS.

Nome della Regione	Codice del comando
Stati Uniti orientali (Ohio)	us-east-2
Stati Uniti orientali (Virginia settentrionale)	us-east-1
Regione Stati Uniti occidentali (California settentrionale)	us-west-1
Stati Uniti occidentali (Oregon)	us-west-2

Nome della Regione	Codice del comando
Regione Canada (Centrale)	ca-central-1
Europa (Irlanda)	eu-west-1
Regione Europa (Londra)	eu-west-2
Regione Europa (Parigi)	eu-west-3
Regione Europa (Francoforte)	eu-central-1
Asia Pacifico (Tokyo)	ap-northeast-1
Asia Pacifico (Seoul)	ap-northeast-2
Regione Asia Pacifico (Mumbai)	ap-south-1
Asia Pacifico (Singapore)	ap-southeast-1
Asia Pacifico (Sydney)	ap-southeast-2
Sud America (San Paolo)	sa-east-1

Per usare un comando CLI, devi disporre delle autorizzazioni appropriate. Per ulteriori informazioni sulle autorizzazioni di AWS OpsWorks Stacks, consulta Gestione delle autorizzazioni utente. Per determinare le autorizzazioni necessarie per un particolare comando, consulta la pagina di riferimento del comando nella guida di riferimento di AWS OpsWorks Stacks.

Le sezioni seguenti descrivono come utilizzare la CLI AWS OpsWorks Stacks per eseguire una serie di attività comuni.

Creare un'istanza (create-instance)

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando create-instance per creare un'istanza in un determinato stack.

Argomenti

- Creare un'istanza con un nome host predefinito
- Creare un'istanza con un nome host a tema
- Creare un'istanza con un'AMI personalizzata

Creare un'istanza con un nome host predefinito

```
C:\>aws opsworks --region us-west-1 create-instance --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
    --layer-ids 5c8c272a-f2d5-42e3-8245-5bf3927cb65b --instance-type m1.large --os
 "Amazon Linux"
```

Gli argomenti sono i seguenti:

- `stack-id`— Puoi ottenere l'ID dello stack dalla pagina delle impostazioni dello stack sulla console (cerca OpsWorks ID) o chiamando describe-stacks.
- `layer-ids`— Puoi ottenere il layer IDs dalla pagina dei dettagli del layer sulla console (cerca OpsWorks ID) o chiamando describe-layers. In questo esempio, l'istanza appartiene a un solo livello.
- `instance-type` - La specifica che definisce la memoria, la CPU, la capacità di storage e il costo orario per l'istanza (`m1.large` in questo esempio).
- `os` - Il sistema operativo dell'istanza; Amazon Linux in questo esempio.

Il comando restituisce un oggetto JSON che contiene l'ID istanza, come segue:

```
{
    "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

In questo esempio viene creata un'istanza con un nome host predefinito, che è semplicemente un numero intero. La sezione seguente descrive come creare un'istanza con un nome host generato da un tema.

Creare un'istanza con un nome host a tema

Puoi anche creare un'istanza con un nome host a tema. Specifica il tema quando crei lo stack. Per maggiori informazioni, consulta Creare un nuovo stack .Per creare l'istanza, prima chiama get-hostname-suggestionper generare un nome. Per esempio:

```
C:\>aws opsworks get-hostname-suggestion --region us-west-1 --layer-id 5c8c272a-
f2d5-42e3-8245-5bf3927cb65b
```

Se specifichi il tema `Layer Dependent` predefinito, `get-hostname-suggestion` aggiunge semplicemente una cifra al nome breve del livello. Per ulteriori informazioni, consulta Creare un nuovo stack.

Il comando restituisce il nome host generato.

```
{
    "Hostname": "php-app2",
    "LayerId": "5c8c272a-f2d5-42e3-8245-5bf3927cb65b"
}
```

Puoi quindi utilizzare l'argomento `hostname` per passare il nome generato a `create-instance`, come segue:

```
c:\>aws --region us-west-1 opsworks create-instance --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
   --layer-ids 5c8c272a-f2d5-42e3-8245-5bf3927cb65b --instance-type m1.large --os
 "Amazon Linux" --hostname "php-app2"
```

Creare un'istanza con un'AMI personalizzata

Il seguente comando create-instance crea un'istanza con un'AMI personalizzata, che deve provenire dalla regione dello stack. Per ulteriori informazioni su come creare un'AMI personalizzata per AWS OpsWorks Stacks, consultaUtilizzo di Custom AMIs.

```
C:\>aws opsworks create-instance --region us-west-1 --stack-id c5ef46ce-3ccd-472c-
a3de-9bec94c6028e
   --layer-ids 6ff8a2ac-c9cc-49cf-9c67-fc852539ade4 --instance-type c3.large --os
 Custom
   --ami-id ami-6c61f104
```

Gli argomenti sono i seguenti:

- `stack-id`— Puoi ottenere l'ID dello stack dalla pagina delle impostazioni dello stack sulla console (cerca OpsWorks ID) o chiamando describe-stacks.
- `layer-ids`— Puoi ottenere il layer IDs dalla pagina dei dettagli del layer sulla console (cerca OpsWorks ID) o chiamando describe-layers. In questo esempio, l'istanza appartiene a un solo livello.
- `instance-type` - Il valore definisce la memoria dell'istanza, la CPU, la capacità di storage e il costo orario e deve essere compatibile con l'AMI (`c3.large` in questo esempio).
- `os` - Il sistema operativo dell'istanza, che deve essere impostato su `Custom` per un'AMI personalizzata.
- `ami-id` - L'ID dell'AMI, che deve avere un aspetto simile a `ami-6c61f104`

> **Note**
>
> Quando utilizzi un'AMI personalizzata, le mappature dei dispositivi a blocchi non sono supportate e i valori specificati per l'opzione `--block-device-mappings` vengono ignorati.

Il comando restituisce un oggetto JSON che contiene l'ID istanza, come segue:

```
{
    "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

Distribuire un'app (create-deployment)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando create-deployment per distribuire un'app su un determinato stack.

Argomenti

- Distribuire un'app

Distribuire un'app

```
aws opsworks --region us-west-1 create-deployment --stack-id cfb7e082-ad1d-4599-8e81-
de1c39ab45bf
   --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb --command "{\"Name\":\"deploy\"}"
```

Gli argomenti sono i seguenti:

- `stack-id`— Puoi ottenere l'ID dello stack dalla pagina delle impostazioni dello stack sulla console (cerca l'OpsWorks ID) o chiamando. `describe-stacks`
- `app-id`— Puoi ottenere l'ID dell'app dalla pagina dei dettagli dell'app (cerca l'OpsWorks ID) o chiamando describe-apps.
- `command` - L'argomento richiede un oggetto JSON che imposta il nome del comando su `deploy`, che distribuisce l'app specificata nello stack.

  I caratteri " nell'oggetto JSON necessitano tutti di caratteri di escape. In caso contrario, il comando potrebbe restituire un errore secondo il quale il JSON non è valido.

Il comando restituisce un oggetto JSON che contiene l'ID di distribuzione, come segue:

```
{
    "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"
}
```

> **Note**
>
> L'esempio precedente esegue la distribuzione in ogni istanza nello stack. Per eseguire la distribuzione su un sottoinsieme di istanze specificato, aggiungi un argomento ed elenca l'istanza. `instance-ids` IDs

Elencare le app di uno stack (describe-apps)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando describe-apps per elencare le app di uno stack o ottenere dettagli sulle app specificate.

```
aws opsworks --region us-west-1 describe-apps --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

L'esempio precedente restituisce un oggetto JSON che contiene informazioni su ogni app. Questo esempio ha una sola app. Per una descrizione di ciascun parametro, consulta describe-apps.

```
{
  "Apps": [
    {
      "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
      "AppSource": {
        "Url": "url",
        "Type": "archive"
      },
      "Name": "SimpleJSP",
      "EnableSsl": false,
      "SslConfiguration": {},
```

```
      "AppId": "da1decc1-0dff-43ea-ad7c-bb667cd87c8b",
      "Attributes": {
        "RailsEnv": null,
        "AutoBundleOnDeploy": "true",
        "DocumentRoot": "ROOT"
      },
      "Shortname": "simplejsp",
      "Type": "other",
      "CreatedAt": "2013-08-01T21:46:54+00:00"
    }
  ]
}
```

Elencare i comandi di uno stack (describe-commands)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando describe-commands per elencare i comandi di uno stack o ottenere dettagli sui comandi specificati. L'esempio seguente ottiene informazioni sui comandi eseguiti su una determinata istanza.

```
aws opsworks --region us-west-1 describe-commands --instance-id
 8c2673b9-3fe5-420d-9cfa-78d875ee7687
```

Il comando restituisce un oggetto JSON che contiene dettagli su ogni comando. Il parametro Type identifica il nome del comando, deploy o undeploy in questo esempio. Per una descrizione degli altri parametri, consulta describe-commands.

```
{
  "Commands": [
    {
```

```
      "Status": "successful",
      "CompletedAt": "2013-07-25T18:57:47+00:00",
      "InstanceId": "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
      "DeploymentId": "6ed0df4c-9ef7-4812-8dac-d54a05be1029",
      "AcknowledgedAt": "2013-07-25T18:57:41+00:00",
      "LogUrl": "https://s3.amazonaws.com/prod_stage-log/logs/008c1a91-
ec59-4d51-971d-3adff54b00cc?AWSAccessKeyId=AIDACKCEVSQ6C2EXAMPLE
 &Expires=1375394373&Signature=HkXil6UuNfxTCC37EPQAa462E1E%3D&response-cache-
control=private&response-content-encoding=gzip&response-content- type=text%2Fplain",
      "Type": "undeploy",
      "CommandId": "008c1a91-ec59-4d51-971d-3adff54b00cc",
      "CreatedAt": "2013-07-25T18:57:34+00:00",
      "ExitCode": 0
    },
    {
      "Status": "successful",
      "CompletedAt": "2013-07-25T18:55:40+00:00",
      "InstanceId": "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
      "DeploymentId": "19d3121e-d949-4ff2-9f9d-94eac087862a",
      "AcknowledgedAt": "2013-07-25T18:55:32+00:00",
      "LogUrl": "https://s3.amazonaws.com/prod_stage-log/
logs/899d3d64-0384-47b6-a586-33433aad117c?AWSAccessKeyId=AIDACKCEVSQ6C2EXAMPLE
 &Expires=1375394373&Signature=xMsJvtLuUqWmsr8s%2FAjVru0BtRs%3D&response-cache-
control=private&response-content-encoding=gzip&response-conten t-type=text%2Fplain",
      "Type": "deploy",
      "CommandId": "899d3d64-0384-47b6-a586-33433aad117c",
      "CreatedAt": "2013-07-25T18:55:29+00:00",
      "ExitCode": 0
    }
  ]
}
```


Elencare le distribuzioni di uno stack (describe-deployments)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando describe-deployments per elencare le distribuzioni di uno stack o ottenere dettagli sulle distribuzioni specificate.

```
aws opsworks --region us-west-1 describe-deployments --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Il comando precedente restituisce un oggetto JSON che contiene dettagli su ogni distribuzione per lo stack specificato. Per una descrizione di ciascun parametro, consulta describe-deployments.


```
{
  "Deployments": [
      {
          "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
          "Status": "successful",
          "CompletedAt": "2013-07-25T18:57:49+00:00",
          "DeploymentId": "6ed0df4c-9ef7-4812-8dac-d54a05be1029",
          "Command": {
              "Args": {},
              "Name": "undeploy"
          },
          "CreatedAt": "2013-07-25T18:57:34+00:00",
          "Duration": 15,
          "InstanceIds": [
              "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
              "9e588a25-35b2-4804-bd43-488f85ebe5b7"
          ]
      },
      {
          "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
          "Status": "successful",
          "CompletedAt": "2013-07-25T18:56:41+00:00",
          "IamUserArn": "arn:aws:iam::444455556666:user/example-user",
          "DeploymentId": "19d3121e-d949-4ff2-9f9d-94eac087862a",
          "Command": {
              "Args": {},
              "Name": "deploy"
          },
          "InstanceIds": [
              "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
              "9e588a25-35b2-4804-bd43-488f85ebe5b7"
          ],
          "Duration": 72,
          "CreatedAt": "2013-07-25T18:55:29+00:00"
```

```
        }
    ]
}
```

Elenca gli indirizzi IP elastici di uno stack () describe-elastic-ips

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando describe-elastic-ips per elencare gli indirizzi IP elastici che sono stati registrati con uno stack o per ottenere dettagli sugli indirizzi IP elastici specificati.

```
aws opsworks --region us-west-2 describe-elastic-ips --instance-id b62f3e04-e9eb-436c-a91f-d9e9a396b7b0
```

Il comando precedente restituisce un oggetto JSON che contiene dettagli su ogni indirizzo IP elastico (uno in questo esempio) per un'istanza specificata. Per una descrizione di ciascun parametro, consulta describe-elastic-ips.

```
{
  "ElasticIps": [
      {
          "Ip": "192.0.2.0",
          "Domain": "standard",
          "Region": "us-west-2"
      }
  ]
}
```

Elencare le istanze di uno stack (describe-instances)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando describe-instances per elencare le istanze di uno stack o per ottenere dettagli sulle istanze specificate.

```
C:\>aws opsworks --region us-west-2 describe-instances --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Il comando precedente restituisce un oggetto JSON che contiene dettagli su ogni istanza nello stack specificato. Per una descrizione di ciascun parametro, consulta describe-instances.

```
{
  "Instances": [
      {
          "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
          "SshHostRsaKeyFingerprint":
 "f4:3b:8e:27:1b:73:98:80:5d:d7:33:e2:b8:c8:8f:de",
          "Status": "stopped",
          "AvailabilityZone": "us-west-2a",
          "SshHostDsaKeyFingerprint":
 "e8:9b:c7:02:18:2a:bd:ab:45:89:21:4e:af:0b:07:ac",
          "InstanceId": "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
          "Os": "Amazon Linux",
          "Hostname": "db-master1",
          "SecurityGroupIds": [],
          "Architecture": "x86_64",
          "RootDeviceType": "instance-store",
          "LayerIds": [
              "41a20847-d594-4325-8447-171821916b73"
          ],
          "InstanceType": "c1.medium",
```

```
            "CreatedAt": "2013-07-25T18:11:27+00:00"
        },
        {
            "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
            "SshHostRsaKeyFingerprint":
 "ae:3a:85:54:66:f3:ce:98:d9:83:39:1e:10:a9:38:12",
            "Status": "stopped",
            "AvailabilityZone": "us-west-2a",
            "SshHostDsaKeyFingerprint":
 "5b:b9:6f:5b:1c:ec:55:85:f3:45:f1:28:25:1f:de:e4",
            "InstanceId": "9e588a25-35b2-4804-bd43-488f85ebe5b7",
            "Os": "Amazon Linux",
            "Hostname": "tomcustom1",
            "SecurityGroupIds": [],
            "Architecture": "x86_64",
            "RootDeviceType": "instance-store",
            "LayerIds": [
                "e6cbcd29-d223-40fc-8243-2eb213377440"
            ],
            "InstanceType": "c1.medium",
            "CreatedAt": "2013-07-25T18:15:52+00:00"
        }
    ]
}
```

Elencare gli stack di un account (describe-stacks)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando describe-stacks per elencare gli stack di un account o per ottenere dettagli sugli stack specificati.

```
aws opsworks --region us-west-2 describe-stacks
```

Il comando precedente restituisce un oggetto JSON che contiene dettagli su ogni stack nell'account (due in questo esempio). Per una descrizione di ciascun parametro, consulta describe-stacks.


```
{
    "Stacks": [
        {
            "ServiceRoleArn": "arn:aws:iam::444455556666:role/aws-opsworks-service-
role",
            "StackId": "aeb7523e-7c8b-49d4-b866-03aae9d4fbcb",
            "DefaultRootDeviceType": "instance-store",
            "Name": "TomStack-sd",
            "ConfigurationManager": {
                "Version": "11.4",
                "Name": "Chef"
            },
            "UseCustomCookbooks": true,
            "CustomJson": "{\n  \"tomcat\": {\n    \"base_version\": 7,\n
 \"java_opts\": \"-Djava.awt.headless=true -Xmx256m\"\n  },\n  \
"datasources\": {\n    \"ROOT\": \"jdbc/mydb\"\n  }\n}",
            "Region": "us-west-2",
            "DefaultInstanceProfileArn": "arn:aws:iam::444455556666:instance-profile/
aws-opsworks-ec2-role",
            "CustomCookbooksSource": {
                "Url": "git://github.com/example-repo/tomcustom.git",
                "Type": "git"
            },
            "DefaultAvailabilityZone": "us-west-2a",
            "HostnameTheme": "Layer_Dependent",
            "Attributes": {
                "Color": "rgb(45, 114, 184)"
            },
            "DefaultOs": "Amazon Linux",
            "CreatedAt": "2013-08-01T22:53:42+00:00"
        },
        {
            "ServiceRoleArn": "arn:aws:iam::444455556666:role/aws-opsworks-service-
role",
            "StackId": "40738975-da59-4c5b-9789-3e422f2cf099",
            "DefaultRootDeviceType": "instance-store",
            "Name": "MyStack",
            "ConfigurationManager": {
                "Version": "11.4",
                "Name": "Chef"
```

```
            },
            "UseCustomCookbooks": false,
            "Region": "us-west-2",
            "DefaultInstanceProfileArn": "arn:aws:iam::444455556666:instance-profile/
aws-opsworks-ec2-role",
            "CustomCookbooksSource": {},
            "DefaultAvailabilityZone": "us-west-2a",
            "HostnameTheme": "Layer_Dependent",
            "Attributes": {
                "Color": "rgb(45, 114, 184)"
            },
            "DefaultOs": "Amazon Linux",
            "CreatedAt": "2013-10-25T19:24:30+00:00"
        }
    ]
}
```

Elencare i livelli di uno stack (describe-layers)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando describe-layers per elencare i livelli di uno stack o per ottenere dettagli sui livelli specificati.

```
aws opsworks --region us-west-2 describe-layers --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Il comando precedente restituisce un oggetto JSON che contiene dettagli su ogni livello in uno stack specificato, in questo esempio un livello MySQL e un livello personalizzato. Per una descrizione di ciascun parametro, consulta describe-layers.

```
{
  "Layers": [
```

```
    {
        "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
        "Type": "db-master",
        "DefaultSecurityGroupNames": [
            "AWS-OpsWorks-DB-Master-Server"
        ],
        "Name": "MySQL",
        "Packages": [],
        "DefaultRecipes": {
            "Undeploy": [],
            "Setup": [
                "opsworks_initial_setup",
                "ssh_host_keys",
                "ssh_users",
                "mysql::client",
                "dependencies",
                "ebs",
                "opsworks_ganglia::client",
                "mysql::server",
                "dependencies",
                "deploy::mysql"
            ],
            "Configure": [
                "opsworks_ganglia::configure-client",
                "ssh_users",
                "agent_version",
                "deploy::mysql"
            ],
            "Shutdown": [
                "opsworks_shutdown::default",
                "mysql::stop"
            ],
            "Deploy": [
                "deploy::default",
                "deploy::mysql"
            ]
        },
        "CustomRecipes": {
            "Undeploy": [],
            "Setup": [],
            "Configure": [],
            "Shutdown": [],
            "Deploy": []
        },
```

```
            "EnableAutoHealing": false,
            "LayerId": "41a20847-d594-4325-8447-171821916b73",
            "Attributes": {
                "MysqlRootPasswordUbiquitous": "true",
                "RubygemsVersion": null,
                "RailsStack": null,
                "HaproxyHealthCheckMethod": null,
                "RubyVersion": null,
                "BundlerVersion": null,
                "HaproxyStatsPassword": null,
                "PassengerVersion": null,
                "MemcachedMemory": null,
                "EnableHaproxyStats": null,
                "ManageBundler": null,
                "NodejsVersion": null,
                "HaproxyHealthCheckUrl": null,
                "MysqlRootPassword": "*****FILTERED*****",
                "GangliaPassword": null,
                "GangliaUser": null,
                "HaproxyStatsUrl": null,
                "GangliaUrl": null,
                "HaproxyStatsUser": null
            },
            "Shortname": "db-master",
            "AutoAssignElasticIps": false,
            "CustomSecurityGroupIds": [],
            "CreatedAt": "2013-07-25T18:11:19+00:00",
            "VolumeConfigurations": [
                {
                    "MountPoint": "/vol/mysql",
                    "Size": 10,
                    "NumberOfDisks": 1
                }
            ]
        },
        {
            "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
            "Type": "custom",
            "DefaultSecurityGroupNames": [
                "AWS-OpsWorks-Custom-Server"
            ],
            "Name": "TomCustom",
            "Packages": [],
            "DefaultRecipes": {
```

```
            "Undeploy": [],
            "Setup": [
                "opsworks_initial_setup",
                "ssh_host_keys",
                "ssh_users",
                "mysql::client",
                "dependencies",
                "ebs",
                "opsworks_ganglia::client"
            ],
            "Configure": [
                "opsworks_ganglia::configure-client",
                "ssh_users",
                "agent_version"
            ],
            "Shutdown": [
                "opsworks_shutdown::default"
            ],
            "Deploy": [
                "deploy::default"
            ]
        },
        "CustomRecipes": {
            "Undeploy": [],
            "Setup": [
                "tomcat::setup"
            ],
            "Configure": [
                "tomcat::configure"
            ],
            "Shutdown": [],
            "Deploy": [
                "tomcat::deploy"
            ]
        },
        "EnableAutoHealing": true,
        "LayerId": "e6cbcd29-d223-40fc-8243-2eb213377440",
        "Attributes": {
            "MysqlRootPasswordUbiquitous": null,
            "RubygemsVersion": null,
            "RailsStack": null,
            "HaproxyHealthCheckMethod": null,
            "RubyVersion": null,
            "BundlerVersion": null,
```

```
                "HaproxyStatsPassword": null,
                "PassengerVersion": null,
                "MemcachedMemory": null,
                "EnableHaproxyStats": null,
                "ManageBundler": null,
                "NodejsVersion": null,
                "HaproxyHealthCheckUrl": null,
                "MysqlRootPassword": null,
                "GangliaPassword": null,
                "GangliaUser": null,
                "HaproxyStatsUrl": null,
                "GangliaUrl": null,
                "HaproxyStatsUser": null
            },
            "Shortname": "tomcustom",
            "AutoAssignElasticIps": false,
            "CustomSecurityGroupIds": [],
            "CreatedAt": "2013-07-25T18:12:53+00:00",
            "VolumeConfigurations": []
        }
    ]
}
```

Eseguire una ricetta (create-deployment)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando create-deployment per eseguire comandi dello stack e comandi di distribuzione. Nell'esempio seguente viene eseguito il comando di uno stack per eseguire una ricetta personalizzata su uno stack specificato.

```
aws opsworks --region us-west-1 create-deployment --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
```

```
    --command "{\"Name\":\"execute_recipes\", \"Args\":{\"recipes\":[\"phpapp::appsetup
\"]}}"
```

L'argomento `command` accetta un oggetto JSON formattato come segue:

- `Name` - Specifica il nome del comando. Il comando `execute_recipes` utilizzato per questo esempio esegue una determinata ricetta sulle istanze dello stack.
- `Args` - Specifica un elenco di argomenti e i relativi valori. Questo esempio dispone di un argomento, `recipes`, che è impostato sulla ricetta da eseguire, `phpapp::appsetup`.

I caratteri " nell'oggetto JSON necessitano tutti di caratteri di escape. In caso contrario, il comando potrebbe restituire un errore secondo il quale il JSON non è valido.

Il comando restituisce un ID di distribuzione che puoi utilizzare per identificare il comando per altri comandi dell'interfaccia CLI come `describe-commands`.

```
{
    "DeploymentId": "5cbaa7b9-4e09-4e53-aa1b-314fbd106038"
}
```

Installare dipendenze (create-deployment)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando create-deployment per eseguire comandi dello stack e comandi di distribuzione. L'esempio seguente esegue il comando dello stack `update_dependencies` per aggiornare le dipendenze sulle istanze di uno stack.

```
aws opsworks --region us-west-1 create-deployment --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
```

```
--command "{\"Name\":\"install_dependencies\"}"
```

L'argomento `command` accetta un oggetto JSON con un parametro `Name`, il cui valore specifica il nome del comando, `install_dependencies` per questo esempio. I caratteri " nell'oggetto JSON necessitano tutti di caratteri di escape. In caso contrario, il comando potrebbe restituire un errore secondo il quale il JSON non è valido.

Il comando restituisce un ID di distribuzione che puoi utilizzare per identificare il comando per altri comandi dell'interfaccia CLI come `describe-commands`.

```
{
    "DeploymentId": "aef5b255-8604-4928-81b3-9b0187f962ff"
}
```

Aggiornare la configurazione dello stack (update-stack)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Utilizza il comando update-stack per aggiornare la configurazione di un determinato stack. L'esempio seguente aggiorna uno stack per aggiungere un JSON personalizzato agli attributi di configurazione dello stack.

```
aws opsworks --region us-west-1 update-stack --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
   --custom-json "{\"somekey\":\"somevalue\"}" --service-role-arn
 arn:aws:iam::444455556666:role/aws-opsworks-service-role
```

I caratteri " nell'oggetto JSON necessitano tutti di caratteri di escape. In caso contrario, il comando potrebbe restituire un errore secondo il quale il JSON non è valido.

> **Note**
>
> Nell'esempio viene inoltre specificato un ruolo del servizio per lo stack. Devi impostare `service-role-arn` su un ARN del ruolo del servizio valido o l'azione non verrà eseguita. Non vi è alcun valore predefinito. Se preferisci puoi specificare l'attuale ARN del ruolo del servizio dello stack, ma devi farlo in modo esplicito.

Il comando `update-stack` non restituisce un valore.

Guida al debugging e alla risoluzione dei problemi

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se devi eseguire il debug di una ricetta o risolvere i problemi relativi a un servizio, l'approccio migliore in genere prevede l'esecuzione delle seguenti fasi nell'ordine indicato:

1. Consultare l'argomento Risoluzione dei problemi e debug di alcuni problemi comuni per l'individuazione di un problema specifico.
2. Eseguire una ricerca nel forum di AWS OpsWorks Stacks per vedere se il problema è già stato discusso.

   Il Forum include molti utenti esperti ed è monitorato dal team di AWS OpsWorks Stacks.
3. Per problemi relativi alle ricette, consulta l'argomento Il debug delle ricette.
4. Contatta l'assistenza di AWS OpsWorks Stacks o pubblica il problema sul forum di AWS OpsWorks Stacks.

La sezione seguente fornisce alcune indicazioni utili per il debug delle ricette. La sezione finale descrive il debug e la risoluzione dei problemi più comuni, fornendo le relative soluzioni.

> **Note**
>
> Ogni esecuzione di Chef genera un log, che fornisce una descrizione dettagliata dell'esecuzione e costituisce una risorsa fondamentale per la risoluzione dei problemi. Per specificare il livello di dettaglio del log, aggiungi un'istruzione `Chef::Log.level` a una ricetta personalizzata che specifica il livello di log desiderato. Il valore predefinito è `:info`. L'esempio seguente mostra come impostare il livello di log di Chef su `:debug`, che restituisce la descrizione più dettagliata possibile dell'esecuzione.
>
> ```
> Chef::Log.level = :debug
> ```
>
> Per ulteriori informazioni sulla visualizzazione e sull'interpretazione dei log di Chef, consulta Log di Chef.

Argomenti

- Il debug delle ricette
- Risoluzione dei problemi e debug di alcuni problemi comuni

Il debug delle ricette

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Quando si verifica un evento del ciclo di vita, o quando esegui il comando relativo allo stack delle ricette di esecuzione, AWS OpsWorks Stacks emette un comando per l'agente al fine di inizializzare il comando di sola esecuzione di Chef sulle istanze specificate per eseguire le ricette appropriate, tra cui le tue ricette personalizzate. Questa sezione descrive alcuni modi per eseguire il debug delle ricette non riuscite.

Argomenti

- Accesso a un'istanza non riuscita
- Log di Chef
- Utilizzo della CLI AWS OpsWorks di Stacks Agent

Accesso a un'istanza non riuscita

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Se una ricetta ha esito negativo, l'istanza finirà sullo stato `setup_failed` anziché su quello online. Anche se l'istanza non è online per quanto riguarda AWS OpsWorks Stacks, l' EC2 istanza è in esecuzione ed è spesso utile accedere per risolvere il problema. Ad esempio, puoi verificare se un'applicazione o un libro di ricette personalizzato è installato correttamente. Il supporto integrato di AWS OpsWorks Stacks per l'accesso SSH e RDP è disponibile solo per le istanze nello stato online. Ad ogni modo, se hai assegnato una coppia di chiavi SSH per l'istanza, puoi comunque effettuare l'accesso come segue:

- Istanze Linux: utilizza la chiave privata della coppia di chiavi SSH per accedere con un client SSH di terze parti, come OpenSSH o PuTTY.

  A tale scopo è possibile utilizzare una coppia di EC2 chiavi o la propria coppia di chiavi SSH personale.
- Istanze Windows: utilizza la EC2 chiave privata della coppia di chiavi per recuperare la password dell'amministratore dell'istanza.

  Utilizza quella password per effettuare l'accesso con il tuo client RDP preferito. Per ulteriori informazioni, consulta Accesso come amministratore. Non puoi utilizzare una coppia di chiavi SSH personale per recuperare una password amministratore.

Log di Chef

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

I registri degli chef sono una delle risorse chiave per la risoluzione dei problemi, in particolare per le ricette di debug. AWS OpsWorks Stacks acquisisce il registro Chef per ogni comando e conserva i log dei 30 comandi più recenti di un'istanza. Poiché l'esecuzione è in modalità di debug, il log contiene una descrizione dettagliata dell'esecuzione di Chef, tra cui il testo inviato a `stdout` e `stderror`. Se una ricetta ha esito negativo, il log include la traccia dello stack di Chef.

AWS OpsWorks Stacks offre diversi modi per visualizzare i log di Chef. Una volta che hai acquisito le informazioni di log, puoi utilizzarle per eseguire il debug delle ricette non riuscite.

> **Note**
>
> Puoi visualizzare anche una coda di log specifica, tramite l'utilizzo di SSH per connettersi all'istanza e per l'esecuzione del comando `show_log` della CLI dell'agente. Per ulteriori informazioni, consulta Visualizzazione dei log di Chef.

Argomenti

- Visualizzazione di un log di Chef con la console
- Visualizzazione di un log di Chef tramite la CLI o l'API
- Visualizzazione di un log di Chef su un'istanza
- Interpretazione di un log di Chef
- Errori comuni del log di Chef

Visualizzazione di un log di Chef con la console

Il modo più semplice per visualizzare un log di Chef è quello di accedere alla pagina dei dettagli dell'istanza. La sezione Logs (Log) include una voce per ogni evento e per il comando relativo all'esecuzione delle ricette. Di seguito viene mostrata una sezione Logs (Log) dell'istanza, con i comandi configure (configura) e setup (imposta), che corrispondono agli eventi del ciclo di vita Configura e Imposta.



Fai clic su show (mostra) nella colonna Log del comando appropriato per visualizzare il log di Chef corrispondente. Se si verifica un errore, AWS OpsWorks Stacks apre automaticamente il registro dell'errore, che di solito si trova alla fine del file.

Visualizzazione di un log di Chef tramite la CLI o l'API

Puoi utilizzare il comando AWS OpsWorks Stacks `describe-commands`CLI o `DescribeCommands`l'azione API per visualizzare i log, che sono archiviati su un bucket Amazon S3. Di seguito viene mostrato come utilizzare la CLI per visualizzare uno qualsiasi dell'attuale serie di file di log per una determinata istanza. La procedura per l'utilizzo di `DescribeCommands` è essenzialmente simile.

Per utilizzare gli AWS OpsWorks Stacks per visualizzare i log Chef di un'istanza

1. Apri la pagina dei dettagli dell'istanza e copia il valore dell'OpsWorksID.
2. Utilizza il valore ID per eseguire il comando della CLI `describe-commands` come segue:

   ```
   aws opsworks describe-commands --instance-id 67bf0da2-29ed-4217-990c-d895d51812b9
   ```

   Il comando restituisce un oggetto JSON con un oggetto incorporato per ogni comando che AWS OpsWorks Stacks ha eseguito sull'istanza, con il più recente per primo. Il parametro `Type` contiene il tipo di comando per ogni oggetto incorporato, un comando `configure` e un comando `setup` in questo esempio.

   ```
   ```

```
{
    "Commands": [
        {
            "Status": "successful",
            "CompletedAt": "2013-10-25T19:38:36+00:00",
            "InstanceId": "67bf0da2-29ed-4217-990c-d895d51812b9",
            "AcknowledgedAt": "2013-10-25T19:38:24+00:00",
            "LogUrl": "https://s3.amazonaws.com/prod_stage-log/logs/
b6c402df-5c23-45b2-a707-ad20b9c5ae40?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Expires=1382731518&Signature=YkqS5IZN2P4wixjHwoC3aCMbn5s%3D&response-cache-
control=private&response-content-encoding=gzip&response-content-
type=text%2Fplain",
            "Type": "configure",
            "CommandId": "b6c402df-5c23-45b2-a707-ad20b9c5ae40",
            "CreatedAt": "2013-10-25T19:38:11+00:00",
            "ExitCode": 0
        },
        {
            "Status": "successful",
            "CompletedAt": "2013-10-25T19:31:08+00:00",
            "InstanceId": "67bf0da2-29ed-4217-990c-d895d51812b9",
            "AcknowledgedAt": "2013-10-25T19:29:01+00:00",
            "LogUrl": "https://s3.amazonaws.com/prod_stage-log/logs/2a90e862-
f974-42a6-9342-9a4f03468358?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Expires=1382731518&Signature=cxKYHO8mCCd4MvOyFb6ywebeQtA%3D&response-cache-
control=private&response-content-encoding=gzip&response-content-
type=text%2Fplain",
            "Type": "setup",
            "CommandId": "2a90e862-f974-42a6-9342-9a4f03468358",
            "CreatedAt": "2013-10-25T19:26:01+00:00",
            "ExitCode": 0
        }
    ]
}
```


3. Copia il valore `LogUrl` sul tuo browser per visualizzare il log.

Se l'istanza dispone di pochi comandi, puoi aggiungere parametri su `describe-commands` per filtrare quali comandi sono incluse nell'oggetto della risposta. Per ulteriori informazioni, consulta describe-commands.

Visualizzazione di un log di Chef su un'istanza

> **Note**
>
> Gli argomenti in questa sezione si applicano a Chef 12. Per ulteriori informazioni sulla posizione dei log di Chef per Chef 11.10 e versioni precedenti, consulta la sezione relativa alla risoluzione di problemi di Chef 11.10 e versioni precedenti per Linux.

Istanze Linux

AWS OpsWorks Stacks memorizza i log Chef di ogni istanza nella sua directory. `/var/chef/runs` (Per le istanze Linux, questa directory include anche i contenitori di dati associati, archiviati come file in formato JSON). È necessario disporre di privilegi sudo per accedere a questa directory. Il log per ogni esecuzione si trova in un file chiamato `chef.log`, all'interno della singola sottodirectory di esecuzione.

AWS OpsWorks Stacks archivia i registri interni nella cartella dell'istanza. `/var/log/aws/opsworks` Le informazioni in genere non sono molto utili per la risoluzione dei problemi. Tuttavia, questi log sono utili per l'assistenza di AWS OpsWorks Stacks e potrebbe esserti richiesto di fornirli se riscontri un problema con il servizio. I log di Linux possono anche fornire dati utili per la risoluzione dei problemi.

Istanze Windows

Log dell'Agente

Nelle istanze di Windows, OpsWorks i log vengono archiviati in un `ProgramData` percorso come il seguente. Il numero include un Time stamp.

```
C:\ProgramData\OpsWorksAgent\var\logs\number
```

> **Note**
>
> Per impostazione predefinita, `ProgramData` è una cartella nascosta. Affinché non sia più nascosta, spostati su Folder Options (Opzioni cartella). Su View (Apri), scegli l'opzione per visualizzare i file nascosti.

L'esempio seguente mostra i log dell'agente su un'istanza di Windows.

```
Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         5/24/2015  11:59 PM     127277 command.20150524.txt
-a---         5/25/2015  11:59 PM     546772 command.20150525.txt
-a---         5/26/2015  11:59 PM     551514 command.20150526.txt
-a---         5/27/2015   9:43 PM     495181 command.20150527.txt
-a---         5/24/2015  11:59 PM      24353 keepalive.20150524.txt
-a---         5/25/2015  11:59 PM     106232 keepalive.20150525.txt
-a---         5/26/2015  11:59 PM     106208 keepalive.20150526.txt
-a---         5/27/2015   8:54 PM      92593 keepalive.20150527.txt
-a---         5/24/2015   7:19 PM       3891 service.20150524.txt
-a---         5/27/2015   8:54 PM       1493 service.20150527.txt
-a---         5/24/2015  11:59 PM     112549 wire.20150524.txt
-a---         5/25/2015  11:59 PM     501501 wire.20150525.txt
-a---         5/26/2015  11:59 PM     499640 wire.20150526.txt
-a---         5/27/2015   8:54 PM     436870 wire.20150527.txt
```

Log di Chef

Sulle istanze Windows, i log di Chef sono archiviati in un percorso `ProgramData` come quello di seguito. Il numero include un Time stamp.

```
C:\ProgramData\OpsWorksAgent\var\commands\number
```

> **Note**
>
> Questa directory contiene solo l'output della prima esecuzione di Chef (OpsWorks di proprietà).

L'esempio seguente mostra i log Chef OpsWorks di proprietà su un'istanza di Windows.

```
 Mode                 LastWriteTime            Name
 ----                 -------------            ----
 d----         5/24/2015   7:23 PM
 configure-7ecb5f47-7626-439b-877f-5e7cb40ab8be
 d----         5/26/2015   8:30 PM            configure-8e74223b-d15d-4372-aeea-
a87b428ffc2b
 d----         5/24/2015   6:34 PM            configure-
c3980a1c-3d08-46eb-9bae-63514cee194b
```

```
 d----         5/26/2015   8:32 PM            grant_remote_access-70dbf834-1bfa-4fce-
b195-e50e85402f4c
 d----         5/26/2015  10:30 PM            revoke_remote_access-1111fce9-843a-4b27-
b93f-ecc7c5e9e05b
 d----         5/24/2015   7:21 PM            setup-754ec063-8b60-4cd4-
b6d7-0e89d7b7aa78
 d----         5/26/2015   8:27 PM            setup-af5bed36-5afd-4115-
af35-5766f88bc039
 d----         5/24/2015   6:32 PM            setup-d8abeffa-24d4-414b-
bfb1-4ad07319f358
 d----         5/24/2015   7:13 PM            shutdown-c7130435-9b5c-4a95-
be17-6b988fc6cf9a
 d----         5/26/2015   8:25 PM
 sync_remote_users-64c79bdc-1f6f-4517-865b-23d2def4180c
 d----         5/26/2015   8:48 PM
 update_custom_cookbooks-2cc59a94-315b-414d-85eb-2bdea6d76c6a
```

Log di Chef utente

I log per l'esecuzione del tuo Chef si trova nei file chiamati `logfile.txt`, in una cartella chiamata dopo il comando di Chef numerato, come nel seguente diagramma.

C:/chef └── `runs` └── `command-12345` ├── `attribs.json` ├── `client.rb` └── `logfile.txt`

Interpretazione di un log di Chef

L'inizio del log contiene per lo più logging di Chef interni.

```
# Logfile created on Thu Oct 17 17:25:12 +0000 2013 by logger.rb/1.2.6
[2013-10-17T17:25:12+00:00] INFO: *** Chef 11.4.4 ***
[2013-10-17T17:25:13+00:00] DEBUG: Building node object for php-app1.localdomain
[2013-10-17T17:25:13+00:00] DEBUG: Extracting run list from JSON attributes provided on
 command line
[2013-10-17T17:25:13+00:00] INFO: Setting the run_list to
 ["opsworks_custom_cookbooks::load", "opsworks_custom_cookbooks::execute"] from JSON
[2013-10-17T17:25:13+00:00] DEBUG: Applying attributes from json file
[2013-10-17T17:25:13+00:00] DEBUG: Platform is amazon version 2013.03
[2013-10-17T17:25:13+00:00] INFO: Run List is [recipe[opsworks_custom_cookbooks::load],
 recipe[opsworks_custom_cookbooks::execute]]
[2013-10-17T17:25:13+00:00] INFO: Run List expands to [opsworks_custom_cookbooks::load,
 opsworks_custom_cookbooks::execute]
[2013-10-17T17:25:13+00:00] INFO: Starting Chef Run for php-app1.localdomain
[2013-10-17T17:25:13+00:00] INFO: Running start handlers
```

```
[2013-10-17T17:25:13+00:00] INFO: Start handlers complete.
[2013-10-17T17:25:13+00:00] DEBUG: No chefignore file found at /opt/aws/opsworks/
releases/20131015111601_209/cookbooks/chefignore no files will be ignored
[2013-10-17T17:25:13+00:00] DEBUG: Cookbooks to compile: ["gem_support", "packages",
 "opsworks_bundler", "opsworks_rubygems", "ruby", "ruby_enterprise", "dependencies",
 "opsworks_commons", "scm_helper", :opsworks_custom_cookbooks]
[2013-10-17T17:25:13+00:00] DEBUG: Loading cookbook gem_support's library file: /
opt/aws/opsworks/releases/20131015111601_209/cookbooks/gem_support/libraries/
current_gem_version.rb
[2013-10-17T17:25:13+00:00] DEBUG: Loading cookbook packages's library file: /opt/aws/
opsworks/releases/20131015111601_209/cookbooks/packages/libraries/packages.rb
[2013-10-17T17:25:13+00:00] DEBUG: Loading cookbook dependencies's library file: /
opt/aws/opsworks/releases/20131015111601_209/cookbooks/dependencies/libraries/
current_gem_version.rb
[2013-10-17T17:25:13+00:00] DEBUG: Loading cookbook opsworks_commons's library file: /
opt/aws/opsworks/releases/20131015111601_209/cookbooks/opsworks_commons/libraries/
activesupport_blank.rb
[2013-10-17T17:25:13+00:00] DEBUG: Loading cookbook opsworks_commons's library file: /
opt/aws/opsworks/releases/20131015111601_209/cookbooks/opsworks_commons/libraries/
monkey_patch_chefgem_resource.rb
...
```

Questa parte del file è utile principalmente per gli esperti di Chef. Nota che l'elenco di esecuzione include solo due ricette, anche se la maggior parte dei comandi ne implicano molte di più. Queste due ricette gestiscono le attività di caricamento e di esecuzione di tutte le altre ricette predefinite e integrate.

La parte più interessante del file solitamente è alla fine. Se una sessione termina correttamente, potresti visualizzare qualcosa di simile a quanto segue:

```
...
[Tue, 11 Jun 2013 16:00:50 +0000] DEBUG: STDERR:
[Tue, 11 Jun 2013 16:00:50 +0000] DEBUG: ---- End output of /sbin/service mysqld
 restart ----
[Tue, 11 Jun 2013 16:00:50 +0000] DEBUG: Ran /sbin/service mysqld restart returned 0
[Tue, 11 Jun 2013 16:00:50 +0000] INFO: service[mysql]: restarted successfully
[Tue, 11 Jun 2013 16:00:50 +0000] INFO: Chef Run complete in 84.07096 seconds
[Tue, 11 Jun 2013 16:00:50 +0000] INFO: cleaning the checksum cache
[Tue, 11 Jun 2013 16:00:50 +0000] DEBUG: removing unused checksum cache file /var/chef/
cache/checksums/chef-file--tmp-chef-rendered-template20130611-4899-8wef7e-0
[Tue, 11 Jun 2013 16:00:50 +0000] DEBUG: removing unused checksum cache file /var/chef/
cache/checksums/chef-file--tmp-chef-rendered-template20130611-4899-1xpwyb6-0
```

```
[Tue, 11 Jun 2013 16:00:50 +0000] DEBUG: removing unused checksum cache file /var/chef/
cache/checksums/chef-file--etc-monit-conf
[Tue, 11 Jun 2013 16:00:50 +0000] INFO: Running report handlers
[Tue, 11 Jun 2013 16:00:50 +0000] INFO: Report handlers complete
[Tue, 11 Jun 2013 16:00:50 +0000] DEBUG: Exiting
```

> **Note**
>
> Puoi utilizzare la CLI dell'agente per visualizzare la coda di log durante o dopo l'esecuzione. Per ulteriori informazioni, consulta Visualizzazione dei log di Chef.

Se una ricetta ha esito negativo, dovresti cercare un output del livello ERROR, che conterrà un'eccezione seguita da una traccia dello stack di Chef, come mostrato di seguito:

```
...
Please report any problems with the /usr/scripts/mysqlbug script!

[  OK  ]
MySQL Daemon failed to start.
Starting mysqld:  [FAILED]STDERR: 130611 15:07:55 [Warning] The syntax '--log-slow-
queries' is deprecated and will be removed in a future release. Please use '--slow-
query-log'/'--slow-query-log-file' instead.
130611 15:07:56 [Warning] The syntax '--log-slow-queries' is deprecated and will be
 removed in a future release. Please use '--slow-query-log'/'--slow-query-log-file'
 instead.
---- End output of /sbin/service mysqld start ----

/opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/mixin/command.rb:184:in `handle_command_failures'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/mixin/command.rb:131:in `run_command'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/provider/service/init.rb:37:in `start_service'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/provider/service.rb:60:in `action_start'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource.rb:406:in `send'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource.rb:406:in `run_action'
```

```
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/runner.rb:53:in `run_action'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/runner.rb:89:in `converge'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/runner.rb:89:in `each'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/runner.rb:89:in `converge'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection.rb:94:in `execute_each_resource'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection/stepable_iterator.rb:116:in `call'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection/stepable_iterator.rb:116:in
 `call_iterator_block'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection/stepable_iterator.rb:85:in `step'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection/stepable_iterator.rb:104:in `iterate'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection/stepable_iterator.rb:55:in
 `each_with_index'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection.rb:92:in `execute_each_resource'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/runner.rb:84:in `converge'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/client.rb:268:in `converge'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/client.rb:158:in `run'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/application/solo.rb:190:in `run_application'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/application/solo.rb:181:in `loop'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/application/solo.rb:181:in `run_application'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/application.rb:62:in `run'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/chef-solo:25
  /opt/aws/opsworks/current/bin/chef-solo:16:in `load'
  /opt/aws/opsworks/current/bin/chef-solo:16
```

La fine del file è la traccia dello stack di Chef. Dovresti anche esaminare l'output appena prima dell'eccezione, perché spesso contiene un errore di sistema, ad esempio `package not available`, che può essere utile anche per determinare la causa dell'errore. In questo caso, il daemon MySQL ha esito negativo durante l'avvio.

Errori comuni del log di Chef

Di seguito sono elencati alcuni errori comuni del log di Chef e come risolverli.

Log non trovato

All'inizio di un'esecuzione di Chef, le istanze ricevono un URL Amazon S3 predefinito che consente di visualizzare il registro su una pagina Web al termine dell'esecuzione di Chef. Poiché questo URL scade dopo due ore, non viene caricato alcun registro sul sito Amazon S3 se un'esecuzione di Chef richiede più di due ore, anche se non si sono verificati problemi durante l'esecuzione di Chef. Il comando per creare un log va a buon fine, ma il log può essere visualizzato solo sull'istanza, non sull'URL prefirmata.

Il log termina bruscamente

Se un log di Chef termina bruscamente senza indicare né il successo né le informazioni di errore, probabilmente hai riscontrato uno stato di bassa memoria che ha impedito a Chef il completamento del log. La soluzione migliore è di riprovare con un'istanza di dimensioni maggiori.

Ricetta o libro di ricette mancanti

Se l'esecuzione di Chef incontra una ricetta o un libro di ricette che non è nella cache del libro di ricette, visualizzerai ciò che segue:

```
DEBUG: Loading Recipe mycookbook::myrecipe via include_recipe
ERROR: Caught exception during execution of custom recipe: mycookbook::myrecipe:
   Cannot find a cookbook named mycookbook; did you forget to add metadata to a
 cookbook?
```

Questa voce indica che il libro di ricette `mycookbook` non è nella cache del libro di ricette. Con Chef 11.4, puoi riscontrare questo errore anche se non dichiari correttamente le dipendenze su `metadata.rb`.

AWS OpsWorks Stacks esegue le ricette dalla cache dei libri di cucina dell'istanza. Scarica i libri di ricette dal tuo archivio per questa cache quando si avvia l'istanza. Tuttavia, AWS OpsWorks Stacks non aggiorna automaticamente la cache su un'istanza online se successivamente

modifichi i libri di cucina nel tuo repository. Se hai modificato i tuoi libri di ricette o ne hai aggiunto di nuovi all'avvio dell'istanza, esegui i seguenti passaggi:

1. Assicurati di aver completato le modifiche nell'archivio.
2. Esegui il comando dello stack Update Cookbooks per aggiornare la cache del libro di ricette con la versione più recente dall'archivio.

Errore del comando locale

Se una risorsa `execute` di Chef ha esito negativo durante l'esecuzione del comando specificato, visualizzerai ciò che segue:

```
DEBUG: ---- End output of ./configure --with-config-file-path=/ returned 2
ERROR: execute[PHP: ./configure] (/root/opsworks-agent/site-cookbooks/php-fpm/
recipes/install.rb line 48) had an error:
   ./configure --with-config-file-path=/
```

Scorri nel log per visualizzare l'output `stderr` e `stdout` del comando, che dovrebbe aiutarti nel determinare perché il comando ha avuto esito negativo.

Errore del pacchetto

Se l'installazione di un pacchetto ha esito negativo, visualizzerai ciò che segue:

```
ERROR: package[zend-server-ce-php-5.3] (/root/opsworks-agent/site-cookbooks/
zend_server/recipes/install.rb line 20)
   had an error: apt-get -q -y --force-yes install zend-server-ce-php-5.3=5.0.4+b17
 returned 100, expected 0
```

Scorri nel log per visualizzare l'output STDOUT e STDERROR del comando, che dovrebbe aiutarti nel determinare perché l'installazione del pacchetto ha avuto esito negativo.

Utilizzo della CLI AWS OpsWorks di Stacks Agent

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande

sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

> **Note**
>
> La CLI dell'agente è disponibile solo per le istanze Linux.

Su ogni istanza online, AWS OpsWorks Stacks installa un agente che comunica con il servizio. Il servizio AWS OpsWorks Stacks a sua volta invia comandi all'agente per eseguire attività come l'avvio di Chef sull'istanza quando si verifica un evento del ciclo di vita. Sulle istanze Linux, l'agente espone un'interfaccia a riga di comando (CLI) che è molto utile per la risoluzione dei problemi. Per eseguire un comando della CLI dell'agente, utilizza SSH per connetterti a un'istanza. Puoi quindi eseguire il comando della CLI dell'agente per eseguire un'ampia gamma di attività, tra cui le seguenti:

- Eseguire ricette.
- Visualizzare i log di Chef.
- Visualizzare JSON di configurazione e distribuzione dello stack.

Per ulteriori informazioni su come impostare una connessione SSH per un'istanza, consulta Accesso con SSH. Inoltre, devi disporre delle autorizzazioni SSH e di quelle sudo per lo stack.

Questa sezione descrive come utilizzare la CLI dell'agente per la risoluzione dei problemi. Per ulteriori informazioni e per un riferimento del comando completo, consulta AWS OpsWorks CLI di Stacks Agent.

Argomenti

- Esecuzione di ricette
- Visualizzazione dei log di Chef
- Visualizzazione del JSON di configurazione e distribuzione dello stack.

Esecuzione di ricette

Il comando `run_command` della CLI dell'agente indica all'agente di rieseguire un comando lanciato in precedenza. I comandi più utili per la risoluzione dei problemi, `setup`, `configure`, `deploy` e

`undeploy` corrispondono ognuno a un evento del ciclo di vita. Questi comandi indicano all'agente di inizializzare l'esecuzione di Chef per eseguire le ricette associate.

> **Note**
>
> Il comando `run_command` è limitato all'esecuzione del gruppo di ricette associato a un comando specificato, di solito le ricette associate a un evento del ciclo di vita. Non puoi utilizzarlo per eseguire una determinata ricetta. Per eseguire una o più ricette specifiche, utilizza il comando dello stack Execute Recipes o operazioni API o CLI equivalenti (`create-deployment` e `CreateDeployment`).

Il comando `run_command` è molto utile per il debugging di ricette personalizzate, soprattutto le ricette assegnate agli eventi del ciclo di vita Configura e Imposta, che non possono essere attivati direttamente dalla console. Utilizzando il comando `run_command`, puoi eseguire quando vuoi una ricetta di un determinato evento, affinché non sia necessario avviare o interrompere le istanze.

> **Note**
>
> AWS OpsWorks Stacks esegue le ricette dalla cache dei libri di cucina dell'istanza, non dall'archivio dei libri di cucina. AWS OpsWorks Stacks scarica i libri di cucina in questa cache all'avvio dell'istanza, ma non aggiorna automaticamente la cache sulle istanze online se successivamente modifichi i tuoi libri di cucina. Se hai modificato i tuoi libri di cucina all'avvio dell'istanza, assicurati di eseguire il comando dello stack relativo all'aggiornamento dei libri di ricette, per aggiornare la cache del libro di ricette alla versione più recente dell'archivio.

L'agente memorizza nella cache solo i comandi più recenti. Puoi elencarli tramite l'esecuzione del comando `list_commands`, che restituisce un elenco di comandi memorizzati nella cache e il momento in cui sono stati eseguiti.

```
sudo opsworks-agent-cli list_commands
2013-02-26T19:08:26        setup
2013-02-26T19:12:01        configure
2013-02-26T19:12:05        configure
2013-02-26T19:22:12        deploy
```

Per eseguire nuovamente il comando più recente, eseguire questa operazione:

```
sudo opsworks-agent-cli run_command
```

Per rieseguire l'istanza più recente di un comando specificato, eseguire questa operazione:

```
sudo opsworks-agent-cli run_command command
```

Ad esempio, per rieseguire le ricette Installazione, puoi eseguire il comando seguente:

```
sudo opsworks-agent-cli run_command setup
```

Ogni comando possiede un JSON di configurazione e distribuzione dello stack che rappresenta lo stato dello stack e della distribuzione al momento dell'esecuzione del comando. Dal momento che i dati possono cambiare tra un comando e il successivo, una vecchia istanza di un comando potrebbe impiegare dei dati differente rispetto a quelli più recenti. Per rieseguire una determinata istanza di un comando, copia il tempo dall'output `list_commands` ed esegui l'operazione seguente:

```
sudo opsworks-agent-cli run_command time
```

I precedenti esempi rieseguono tutti il comando utilizzando il JSON predefinito, che è il JSON installato per quel comando. Puoi eseguire nuovamente un comando contro un file JSON arbitrario eseguendo l'operazione illustrata di seguito:

```
sudo opsworks-agent-cli run_command -f /path/to/valid/json.file
```

Visualizzazione dei log di Chef

Il commando `show_log` della CLI dell'agente mostra un log specifico. Al termine del comando, lo vedrai alla fine del file. Il comando `show_log` pertanto offre un modo conveniente per accodare il log, che in genere è la posizione delle informazioni di errore. Puoi scorrere fino a visualizzare le parti precedenti del log.

Per visualizzare il log del comando corrente, eseguire questa operazione:

```
sudo opsworks-agent-cli show_log
```

Puoi anche visualizzare i log per un determinato comando, ma i log vengono memorizzati nella cache dell'agente solo per gli ultimi trenta comandi. Puoi elencare i comandi di un'istanza eseguendo

`list_commands`, che restituisce un elenco dei comandi memorizzati nella cache e il tempo impiegato per eseguirli. Per vedere un esempio, consulta Esecuzione di ricette.

Per visualizzare il log per la più recente l'esecuzione di un determinato comando, eseguire l'operazione seguente:

```
sudo opsworks-agent-cli show_log command
```

Il parametro del comando può essere impostato su `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop`, o `restart`. La maggior parte di questi comandi corrispondono agli eventi del ciclo di vita e indirizzano l'agente per eseguire le ricette associate.

Per visualizzare il log di un determinato comando di esecuzione, copia la data dall'output `list_commands` ed esegui:

```
sudo opsworks-agent-cli show_log date
```

Se un comando è ancora in esecuzione, `show_log` mostrerà lo stato attuale del log.

> **Note**
>
> Un modo per `show_log` risolvere errori e out-of-memory problemi consiste nel tracciare un registro durante l'esecuzione, nel modo seguente:
>
> 1. Utilizza il comando `run_command` per attivare l'evento del ciclo di vita appropriato. Per ulteriori informazioni, consulta Esecuzione di ricette.
> 2. Esegui ripetutamente `show_log` per visualizzare la coda del log nel momento in cui viene scritta.
>
> Se Chef termina la memoria o esce inaspettatamente, il log terminerà bruscamente. Se una ricetta ha esito negativo, il log terminerà con un'eccezione e una traccia dello stack.

Visualizzazione del JSON di configurazione e distribuzione dello stack.

La maggior parte dei dati utilizzati dalle ricette provengono dal JSON di configurazione e distribuzione dello stack. Questo definisce una serie di attributi di Chef che forniscono una descrizione dettagliata della configurazione dello stack, di tutte le distribuzioni e degli attributi opzionali che può aggiungere

un utente. Per ogni comando, AWS OpsWorks Stacks installa un JSON che rappresenta lo stack e lo stato di distribuzione al momento del comando. Per ulteriori informazioni, consulta Attributi di configurazione e distribuzione dello stack.

Se le tue ricette personalizzate ottengono i dati dalla configurazione dello stack e dalla distribuzione JSON, puoi verificare i dati esaminando il JSON. Il modo più semplice per visualizzare il JSON di configurazione e distribuzione dello stack è di eseguire il comando `get_json` della CLI dell'agente, che consente di visualizzare una versione formattata dell'oggetto JSON. Il seguente esempio mostra le prime righe di alcuni output tipici:

```
{
  "opsworks": {
    "layers": {
      "php-app": {
        "id": "4a2a56c8-f909-4b39-81f8-556536d20648",
        "instances": {
          "php-app2": {
            "elastic_ip": null,
            "region": "us-west-2",
            "booted_at": "2013-02-26T20:41:10+00:00",
            "ip": "10.112.235.192",
            "aws_instance_id": "i-34037f06",
            "availability_zone": "us-west-2a",
            "instance_type": "c1.medium",
            "private_dns_name": "ip-10-252-0-203.us-west-2.compute.internal",
            "private_ip": "10.252.0.203",
            "created_at": "2013-02-26T20:39:39+00:00",
            "status": "online",
            "backends": 8,
            "public_dns_name": "ec2-10-112-235-192.us-west-2.compute.amazonaws.com"
...
```

Puoi visualizzare la configurazione dello stack più recente e la distribuzione JSON come segue:

```
sudo opsworks-agent-cli get_json
```

Puoi visualizzare la configurazione più recente e la distribuzione JSON per un comando specificato eseguendo la seguente operazione:

```
sudo opsworks-agent-cli get_json command
```

Il parametro del comando può essere impostato su `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop`, o `restart`. La maggior parte di questi comandi corrispondono agli eventi del ciclo di vita e indirizzano l'agente per eseguire le ricette associate.

Puoi visualizzare la configurazione dello stack e la distribuzione JSON per l'esecuzione di un determinato comando, specificando la data del comando come nell'esempio che segue:

```
sudo opsworks-agent-cli get_json date
```

Il modo più semplice per l'utilizzo di questo comando è il seguente:

1. Esegui `list_commands`, che restituisce un elenco di comandi che sono stati eseguiti sull'istanza e la data che stava eseguendo ogni comando.
2. Copia la data del comando appropriato e usala come argomento. `get_json date`

Risoluzione dei problemi e debug di alcuni problemi comuni

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione descrive il debug e la risoluzione di alcuni dei problemi più comuni, fornendo le relative soluzioni.

Argomenti

- Risoluzione dei problemi relativi agli stack AWS OpsWorks
- Risoluzione dei problemi di registrazione delle istanze

Risoluzione dei problemi relativi agli stack AWS OpsWorks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Questa sezione contiene alcuni problemi di AWS OpsWorks Stacks più comuni e le relative soluzioni.

Argomenti

- Impossibile gestire le istanze
- Dopo un'esecuzione di Chef le istanze non vengono avviate
- Tutte le istanze di un livello non superano l'Elastic Load Balancing Health Check
- Impossibile comunicare con un Elastic Load Balancing Load Balancer
- Un'istanza in locale di rilievo non completa la configurazione del volume dopo il riavvio
- Un volume EBS non viene ricollegato dopo un riavvio
- Impossibile eliminare i gruppi di sicurezza di AWS OpsWorks Stacks
- Eliminato accidentalmente un gruppo di sicurezza Stacks AWS OpsWorks
- Interruzione imprevista del log di Chef
- Il libro di ricette non viene aggiornato
- Le istanze si bloccano sullo stato Booting (Avvio in corso)
- Riavvio imprevisto delle istanze
- opsworks-agent Processi in esecuzione nelle istanze
- Comandi execute_recipes imprevisti

Impossibile gestire le istanze

Problema: non sei in grado di gestire un'istanza che in passato riuscivi invece a gestire. In alcuni casi, nei log è riportato un errore simile al seguente.

```
Aws::CharlieInstanceService::Errors::UnrecognizedClientException - The security token
 included in the request is invalid.
```

Causa: ciò può essere dovuto alla modifica o all'eliminazione di una risorsa esterna a AWS OpsWorks da cui dipende l'istanza. Di seguito sono elencati alcuni esempi di modifiche di risorse che possono interrompere le comunicazioni con un'istanza.

- Un utente o un ruolo IAM associato all'istanza è stato eliminato accidentalmente, al di fuori di AWS OpsWorks Stacks. Ciò causa un errore di comunicazione tra l' AWS OpsWorks agente installato sull'istanza e il servizio AWS OpsWorks Stacks. L'utente associato a un'istanza è un'informazione obbligatoria per tutta la durata dell'istanza.
- La modifica delle configurazioni relative a volumi o storage quando un'istanza è offline può rendere ingestibile un'istanza.
- Aggiungere EC2 istanze a un ELB manualmente. AWS OpsWorks riconfigura un load balancer Elastic Load Balancing assegnato ogni volta che un'istanza entra o esce dallo stato online. AWS OpsWorks considera solo le istanze che conosce come membri validi; le istanze che vengono aggiunte all'esterno o mediante un altro AWS OpsWorks processo vengono rimosse. Ogni altra istanza viene rimossa.

Soluzione: non eliminare gli utenti o i ruoli IAM da cui dipendono le istanze. Se possibile, modificare le configurazioni relative a volumi o storage solo quando le istanze dipendenti sono in esecuzione. Utilizzalo AWS OpsWorks per gestire il bilanciamento del carico o le appartenenze EIP delle istanze. AWS OpsWorks Quando registri un'istanza, per evitare problemi di gestione delle istanze registrate nel caso in cui l'utente venga eliminato accidentalmente, aggiungi il `--use-instance-profile` parametro al `register` comando per utilizzare invece il profilo di istanza integrato dell'istanza.

Dopo un'esecuzione di Chef le istanze non vengono avviate

Problema: negli stack Chef 11.10 o di versioni precedenti non configurati per l'uso di libri di ricette personalizzati, dopo un'esecuzione di Chef in cui sono stati utilizzati libri di ricette della community, le istanze non vengono avviate. Nei messaggi di log può essere indicato che la compilazione delle ricette ha avuto esito negativo ("Recipe Compile Error" (Errore di compilazione della ricetta)) oppure che risulta impossibile caricare le ricette perché non è stata trovata alcuna dipendenza.

Causa: la causa più comune è che il libri di ricette personalizzato o della community non supporta la versione di Chef utilizzata dallo stack. Alcuni libri di ricette della community più diffusi, ad esempio `apt` e `build-essential`, sono caratterizzati da problemi di compatibilità con Chef 11.10 noti.

Soluzione: negli AWS OpsWorks stack con l'impostazione Usa libri di cucina personalizzati Chef attivata, i ricettari personalizzati o comunitari devono sempre supportare la versione di Chef utilizzata dallo stack. Bloccare i libri di ricette della community a una versione specifica, ovvero impostare il numero di versione dei libri di ricette su una versione specifica che sia compatibile con la versione di Chef configurata nelle impostazioni dello stack. Per trovare una versione di libro di ricette della community supportata, visualizzare i log delle modifiche per un libro di ricette non compilato e utilizzare solo la versione più recente del libro di ricette supportata dallo stack. Per bloccare la versione di un libro di ricette, specificare il numero di versione esatto nel file Berksfile dell'archivio di libri di ricette personalizzati. Ad esempio `cookbook 'build-essential', '= 3.2.0'`.

Tutte le istanze di un livello non superano l'Elastic Load Balancing Health Check

Problema: colleghi un load balancer Elastic Load Balancing a un livello di app server, ma tutte le istanze non superano il controllo di integrità.

Causa: quando crei un load balancer Elastic Load Balancing, devi specificare il percorso ping chiamato dal load balancer per determinare se l'istanza è integra. Assicurarsi di specificare un percorso di ping appropriato per l'applicazione. Il valore predefinito è /index.html. Se l'applicazione non include un file `index.html`, è necessario specificare un percorso appropriato. In caso contrario, il controllo dello stato avrà esito negativo. Ad esempio, l'PHPApp applicazione Simple utilizzata in Nozioni di base sugli stack Linux Chef 11 non utilizza index.html; il percorso ping appropriato per tali server è /.

Soluzione: modificare il percorso di ping del sistema di bilanciamento del carico. Per ulteriori informazioni, consulta Elastic Load Balancing.

Impossibile comunicare con un Elastic Load Balancing Load Balancer

Problema: si crea un load balancer Elastic Load Balancing e lo si collega a un livello di app server, ma quando si fa clic sul nome DNS o sull'indirizzo IP del load balancer per eseguire l'applicazione, viene visualizzato il seguente errore: «Il server remoto non risponde».

Causa: se lo stack è in esecuzione in un VPC predefinito, quando crei un sistema di bilanciamento del carico Elastic Load Balancing nella regione, devi specificare un gruppo di sicurezza. Il gruppo di sicurezza deve avere regole in entrata che consentono il traffico in entrata dall'indirizzo IP in uso. Se si specifica un valore nel campo default VPC security group (gruppo di sicurezza del VPC predefinito), la regola in entrata predefinita non accetta traffico in entrata.

Soluzione: modificare le regole in entrata del gruppo di sicurezza in modo che venga accettato il traffico in entrata dagli indirizzi IP appropriati.

1. Fai clic su Gruppi di sicurezza nel riquadro di navigazione della EC2 console Amazon.
2. Impostare il gruppo di sicurezza del sistema di bilanciamento del carico.
3. Fare clic su Edit (Modifica) nella scheda Inbound (In entrata).
4. Aggiungere una regola in entrata con l'opzione Source (Origine) impostata su un CIDR appropriato.

   Ad esempio, se si specifica Anywhere (Ovunque), il CIDR viene impostato su 0.0.0.0/0, ovvero viene indicato al sistema di bilanciamento del carico di accettare il traffico in entrata da qualsiasi indirizzo IP.

Un'istanza in locale di rilievo non completa la configurazione del volume dopo il riavvio

Problema: riavvii un' EC2 istanza che hai importato in AWS OpsWorks Stacks e la console AWS OpsWorks Stacks visualizza lo stato dell'istanza non riuscito. Ciò si può verificare su istanze Chef 11 o Chef 12.

Causa: AWS OpsWorks Stacks potrebbe non essere in grado di collegare un volume all'istanza durante il processo di configurazione. Una causa possibile è che AWS OpsWorks Stacks sovrascrive la configurazione del volume sull'istanza quando esegue il comando `setup`.

Soluzione: aprire la pagina Details (Dettagli) relativa all'istanza, quindi controllare la configurazione del volume nell'area Volumes (Volumi). Si noti che è possibile modificare la configurazione del volume solo quando lo stato dell'istanza è stopped (arrestato). Assicurarsi che ogni volume disponga di un punto di montaggio e di un nome. Verifica di aver fornito il punto di montaggio corretto nella configurazione in AWS OpsWorks Stacks prima di riavviare l'istanza.

Un volume EBS non viene ricollegato dopo un riavvio

Problema: usi la EC2 console Amazon per collegare un volume Amazon EBS a un'istanza, ma quando riavvii l'istanza, il volume non è più collegato.

Causa: AWS OpsWorks gli stack possono ricollegare solo i volumi Amazon EBS di cui è a conoscenza, limitati a quanto segue:

- Volumi creati da Stacks. AWS OpsWorks
- I volumi dell'account in uso esplicitamente registrati con uno stack utilizzando la pagina Resources (Risorse).

Soluzione: gestisci i tuoi volumi Amazon EBS solo utilizzando la console AWS OpsWorks Stacks, l'API o la CLI. Se desideri utilizzare uno dei volumi Amazon EBS del tuo account con uno stack, utilizza la pagina Risorse dello stack per registrare il volume e collegarlo a un'istanza. Per ulteriori informazioni, consulta Gestione delle risorse.

Impossibile eliminare i gruppi di sicurezza di AWS OpsWorks Stacks

Problema: dopo aver eliminato uno stack, rimangono diversi gruppi di sicurezza AWS OpsWorks Stacks che non possono essere eliminati.

Causa: i gruppi di sicurezza devono essere eliminati in base a un determinato ordine.

Soluzione: accertarsi innanzitutto che nessuna istanza stia utilizzando i gruppi di sicurezza. Eliminare quindi i seguenti gruppi di sicurezza, se esistenti, nell'ordine indicato:

1. AWS- OpsWorks -Server vuoto
2. Server principale di monitoraggio OpsWorks AWS
3. Server principale AWS OpsWorks -DB
4. AWS- Server OpsWorks Memcached
5. AWS- OpsWorks -Server personalizzato
6. AWS- OpsWorks -NodeJS-App-Server
7. Server di app AWS per OpsWorks PHP
8. AWS- OpsWorks -Rails App-Server
9. Server Web OpsWorks AWS
10. AWS- OpsWorks -Server-Default
11. AWS- OpsWorks -LB-Server

Eliminato accidentalmente un gruppo di sicurezza Stacks AWS OpsWorks

Problema: hai eliminato uno dei gruppi di sicurezza AWS OpsWorks Stacks e devi ricrearlo.

Causa: questi gruppi di sicurezza vengono in genere eliminati per errore.

Soluzione: il gruppo ricreato deve essere la copia esatta dell'originale, rispettando anche l'uso di maiuscole e minuscole per il nome del gruppo. Anziché ricreare il gruppo manualmente, ti consigliamo di far eseguire l'attività automaticamente da AWS OpsWorks Stacks. Basta creare un nuovo stack nella stessa regione AWS e il VPC, se AWS OpsWorks presente, e Stacks ricreerà

automaticamente tutti i gruppi di sicurezza integrati, incluso quello che hai eliminato. È quindi possibile eliminare lo stack se non è più necessario. I gruppi di sicurezza verranno conservati.

Interruzione imprevista del log di Chef

Problema: Un log di Chef viene interrotto in modo imprevisto. Ciò non indica una corretta esecuzione, né visualizza un'eccezione o una traccia di stack.

Causa: questo comportamento è in genere causato da memoria insufficiente.

Soluzione: creare un'istanza di dimensioni maggiori e utilizzare il comando `run_command` della CLI dell'agente per eseguire di nuovo le ricette. Per ulteriori informazioni, consulta Esecuzione di ricette.

Il libro di ricette non viene aggiornato

Problema: è stato eseguito l'aggiornamento dei libri di ricette, ma le istanze dello stack continuano a eseguire le vecchie ricette.

Causa: AWS OpsWorks Stacks memorizza nella cache i libri di cucina su ogni istanza ed esegue le ricette dalla cache, non dal repository. Quando avvii una nuova istanza, AWS OpsWorks Stacks scarica i tuoi libri di cucina dal repository nella cache dell'istanza. Tuttavia, se in seguito i libri di ricette personalizzati vengono modificati, AWS OpsWorks Stacks non aggiorna automaticamente le cache delle istanze online.

Soluzione: esegui il comando Update Cookbooks stack per indirizzare esplicitamente a AWS OpsWorks Stacks l'aggiornamento delle cache dei libri di cucina delle tue istanze online.

Le istanze si bloccano sullo stato Booting (Avvio in corso)

Problema: quando si riavvia un'istanza oppure quando la diagnostica automatica la riavvia automaticamente, l'operazione di avvio si blocca sullo stato `booting`.

Causa: una possibile causa di questo problema è la configurazione del VPC, compresa la configurazione di un VPC predefinito. Le istanze devono essere sempre in grado di comunicare con il servizio AWS OpsWorks Stacks, Amazon S3 e gli archivi di pacchetti, libri di cucina e app. Se, ad esempio, rimuovi un gateway predefinito da un VPC predefinito, le istanze perdono la connessione al AWS OpsWorks servizio Stacks. Poiché AWS OpsWorks Stacks non può più comunicare con l'agent dell'istanza, considera l'istanza come fallita e la guarisce automaticamente. Tuttavia, senza una connessione, AWS OpsWorks Stacks non può installare un agent di istanza sull'istanza riparata. Senza un agente, AWS OpsWorks Stacks non può eseguire le ricette di installazione sull'istanza, quindi l'operazione di avvio non può andare oltre lo stato di «avvio».

Soluzione: modificare la configurazione del VPC in modo che le istanze dispongano della connettività richiesta.

Riavvio imprevisto delle istanze

Problema: un'istanza arrestata viene riavviata inaspettatamente.

Causa 1: se hai abilitato la riparazione automatica per le tue istanze, AWS OpsWorks Stacks esegue periodicamente un controllo dello stato delle EC2 istanze Amazon associate e riavvia quelle non integre. Se interrompi o interrompi un'istanza AWS OpsWorks gestita da Stacks utilizzando la EC2 console Amazon, l'API o la CLI, Stacks non riceve alcuna notifica. AWS OpsWorks Considererà invece l'istanza arrestata come un'istanza non integra e pertanto la riavvierà automaticamente.

Soluzione: gestire le istanze solo utilizzando la console AWS OpsWorks Stacks, l'API o la CLI. Se usi AWS OpsWorks Stacks per interrompere o eliminare un'istanza, questa non verrà riavviata. Per ulteriori informazioni, consulta Avvio, arresto e riavvio manuali delle istanze 24 ore su 24, 7 giorni su 7 e Eliminazione delle AWS OpsWorks istanze di Stacks.

Causa 2: le istanze possono avere esito negativo per una serie di motivi. Se hai abilitato la riparazione automatica, AWS OpsWorks Stacks riavvia automaticamente le istanze non riuscite.

Soluzione: si tratta di un'operazione normale; non è necessario fare nulla a meno che non si desideri che AWS OpsWorks Stacks riavvii le istanze non riuscite. In questo caso, è necessario disabilitare la diagnostica automatica.

`opsworks-agent` Processi in esecuzione nelle istanze

Problema: sulle istanze sono in esecuzione diversi processi `opsworks-agent`. Per esempio:

```
aws 24543 0.0 1.3 172360 53332 ? S Feb24 0:29 opsworks-agent: master 24543
aws 24545 0.1 2.0 208932 79224 ? S Feb24 22:02 opsworks-agent: keep_alive of master
 24543
aws 24557 0.0 2.0 209012 79412 ? S Feb24 8:04 opsworks-agent: statistics of master
 24543
aws 24559 0.0 2.2 216604 86992 ? S Feb24 4:14 opsworks-agent: process_command of master
 24
```

Causa: si tratta di processi validi necessari per il normale funzionamento dell'agente. Vengono eseguite attività quali la gestione delle distribuzioni e l'invio di messaggi keep-alive al servizio.

Soluzione: questo comportamento è normale. Non arrestare questi processi per evitare di compromettere il funzionamento dell'agente.

Comandi execute_recipes imprevisti

Problema: la sezione Logs (Log) nella pagina dei dettagli di un'istanza include comandi `execute_recipes` imprevisti. I comandi `execute_recipes` imprevisti possono venire visualizzati anche nelle pagine Stack e Deployments (Distribuzioni).

Causa: questo problema è spesso causato da modifiche apportate alle autorizzazioni. Quando modifichi le autorizzazioni SSH o sudo di un utente o di un gruppo, AWS OpsWorks Stacks viene eseguito `execute_recipes` per aggiornare le istanze dello stack e attiva anche un evento Configure. Un'altra origine del comando `execute_recipes` è l'aggiornamento da parte di AWS OpsWorks Stacks dell'agente dell'istanza.

Soluzione: si tratta di un'operazione normale. Non è necessario eseguire alcuna azione.

Per controllare le operazioni eseguite da un comando `execute_recipes`, passare alla pagina Deployments (Distribuzioni) e fare clic sul timestamp del comando. Viene visualizzata la pagina dei dettagli del comando, dove sono elencate le ricette chiave eseguite. Ad esempio, la seguente pagina dei dettagli fa riferimento a un comando `execute_recipes` che ha eseguito `ssh_users` per aggiornare le autorizzazioni SSH.



Per visualizzare tutti i dettagli, fare clic su show (mostra) nella colonna Log del comando per visualizzare il log di Chef associato. Cerca nel registro. **`Run List`** AWS OpsWorks Le ricette per la manutenzione delle pile saranno disponibili in `OpsWorks Custom Run List` basso. Ad esempio, di seguito è riportato l'elenco di esecuzioni per il comando `execute_recipes` visualizzato nel precedente screenshot e mostra ogni ricetta associata al comando.

```
[2014-02-21T17:16:30+00:00] INFO: OpsWorks Custom Run List:
 ["opsworks_stack_state_sync",
  "ssh_users", "test_suite", "opsworks_cleanup"]
```

Risoluzione dei problemi di registrazione delle istanze

In questa sezione sono descritti alcuni dei problemi di registrazione delle istanze più comunemente riscontrati e le loro soluzioni.

> **Note**
>
> Se si sono verificati problemi di registrazione, eseguire `register` con l'argomento `--debug`, che fornisce ulteriori informazioni di debug.

Argomenti

- EC2L'utente non è autorizzato a eseguire:...
- L'ambito delle credenziali deve fare riferimento a una regione valida

EC2L'utente non è autorizzato a eseguire:...

Problema: un comando `register` restituisce il seguente messaggio:

```
A client error (AccessDenied) occurred when calling the CreateGroup operation:
User: arn:aws:iam::123456789012:user/ImportEC2User is not authorized to
perform: iam:CreateGroup on resource:
arn:aws:iam::123456789012:group/AWS/OpsWorks/OpsWorks-b583ce55-1d01-4695-b3e5-
ee19257d1911
```

Causa: il `register` comando viene eseguito con credenziali che non concedono le autorizzazioni richieste. La policy dell'utente deve consentire, tra le altre, l'operazione `iam:CreateGroup`.

Soluzione: fornire al comando `register` le credenziali dell'utente IAM con le autorizzazioni richieste. Per ulteriori informazioni, consulta Installazione e configurazione dell' AWS CLI.

L'ambito delle credenziali deve fare riferimento a una regione valida

Problema: un comando `register` restituisce il seguente messaggio:

```
A client error (InvalidSignatureException) occurred when calling the
DescribeStacks operation: Credential should be scoped to a valid region, not 'cn-
north-1'.
```

Causa: la regione del comando deve essere una regione di AWS OpsWorks Stacks valida. Per un elenco delle regioni supportate, consulta Supporto nelle regioni. Questo errore in genere si verifica per uno dei motivi seguenti:

- Lo stack si trova in una regione diversa e la regione di uno stack è stata assegnata all'argomento `--region` del comando.

  Non è necessario specificare una regione dello stack; AWS OpsWorks Stacks la determina automaticamente dall'ID dello stack.
- È stato omesso l'argomento `--region`, che specifica implicitamente la regione predefinita, ma la regione predefinita non è supportata da AWS OpsWorks Stacks.

Soluzione: imposta `--region` esplicitamente una regione AWS OpsWorks Stacks supportata o modifica il AWS CLI `config` file per cambiare l'area predefinita in una regione Stacks supportata. AWS OpsWorks Per ulteriori informazioni, consultare la sezione relativa alla configurazione dell'interfaccia a riga di comando AWS.

AWS OpsWorks CLI di Stacks Agent

Important

Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Note

Questa caratteristica è disponibile solo per le istanze Linux.

L'agente che AWS OpsWorks Stacks installa su ogni istanza espone un'interfaccia a riga di comando (CLI). Se utilizzi un SSH per accedere all'istanza, puoi utilizzare la CLI per le seguenti operazioni:

- Accedere ai file di log per l'esecuzione di Chef.
- Accedi ai comandi Stacks. AWS OpsWorks
- Eseguire manualmente ricette di Chef.
- Visualizzare rapporti dell'istanza.
- Visualizzare rapporti dell'agente.
- Visualizzare una serie limitata di configurazioni dello stack e degli attributi di distribuzione.

> **Important**
>
> Puoi eseguire comandi della CLI dell'agente solo come root o mediante l'uso di `sudo`.

La sintassi del comando base è:

```
sudo opsworks-agent-cli [--help] [command [activity] [date]]
```

I quattro argomenti sono come quelli indicati di seguito:

help

(facoltativo) Mostra una breve sintesi dei comandi disponibili quando viene utilizzato da solo. Quando viene utilizzato con un comando, `help` mostra una descrizione del comando.

command

(facoltativo) Il comando della CLI dell'agente, che deve essere impostato su uno dei seguenti valori:

- agent_report
- get_json
- instance_report
- list_commands
- run_command
- show_log

- stack_state

activity

(facoltativo) Viene utilizzato come argomento con alcuni comandi per specificare una determinata attività di AWS OpsWorks Stacks: `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop` o `restart`.

data

(facoltativo) Viene utilizzato come argomento con alcuni comandi per specificare un determinato comando di esecuzione di AWS OpsWorks Stacks. Specificate l'esecuzione del comando impostando la data sul timestamp in cui il comando è stato eseguito nel *`yyyy-mm-ddThh:mm:ss`* formato, incluse le virgolette singole. Ad esempio, per le 10:31:55 di martedì 5 febbraio 2013, utilizza: `'2013-02-05T10:31:55'`. Per determinare quando è stato eseguito un particolare comando AWS OpsWorks Stacks, esegui. list_commands

> **Note**
>
> Se l'agente ha eseguito la stessa attività AWS OpsWorks Stacks più volte, puoi scegliere una particolare esecuzione specificando sia l'attività che l'ora in cui è stata eseguita. Se specifichi un'attività omettendo l'ora, il comando della CLI dell'agente agisce sull'esecuzione più recente dell'attività. Se ometti entrambi gli argomenti, il comando della CLI dell'agente agisce sull'attività più recente.

Le seguenti sezioni descrivono i comandi e i relativi argomenti. Per brevità, le sezioni della sintassi omettono l'opzione facoltativa `--help`, che può essere utilizzata con qualsiasi comando.

Argomenti

- agent_report
- get_json
- instance_report
- list_commands
- run_command
- show_log
- stack_state

agent_report

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Restituisce un report dell'agente.

```
sudo opsworks-agent-cli agent_report
```

Il seguente output di esempio proviene da un'istanza che ha eseguito un'attività di configurazione recentemente.

```
$ sudo opsworks-agent-cli agent_report

AWS OpsWorks Instance Agent State Report:

  Last activity was a "configure" on 2015-12-01 18:19:23 UTC
  Agent Status: The AWS OpsWorks agent is running as PID 30998
  Agent Version: 4004-20151201152533, up to date
```

get_json

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Restituisce informazioni su un'esecuzione di Chef come oggetto JSON.

```
sudo opsworks-agent-cli get_json [activity] [date] [-i | --internal | --no-i | --no-
internal]
```

Per impostazione predefinita, `get_json` mostra le informazioni fornite dal cliente per l'esecuzione più recente di Chef. Utilizza le seguenti opzioni per specificare una determinata serie di informazioni.

activity

Mostra le informazioni per l'esecuzione di Chef associate alla più recente attività specificata. Per ottenere un elenco delle attività valide, esegui `list_commands`.

data

Mostra le informazioni per l'esecuzione di Chef associate all'attività eseguita per un Time stamp specifico. Per ottenere un elenco dei Time stamp validi, esegui `list_commands`.

-i, --internal

Visualizza le informazioni che AWS OpsWorks Stacks utilizza internamente per l'esecuzione di Chef.

--no-i, --no-internal

Mostra esplicitamente le informazioni fornite dal cliente per l'esecuzione di Chef. Se non viene specificato altro, questa è l'impostazione predefinita.

> **Note**
>
> Per le istanze Linux di Chef 12, l'esecuzione di questo comando restituirà delle informazioni valide, ad esempio la configurazione dello stack e gli attributi di configurazione dell'istanza. Tuttavia, per ottenere informazioni più complete, fai riferimento ai data bag Chef che AWS OpsWorks Stacks crea sull'istanza. Per ulteriori informazioni, consulta la AWS OpsWorks Riferimento a Stacks Data Bag.

Il seguente output di esempio illustra le informazioni fornite dal cliente per l'esecuzione di più recente di Chef per le attività di configurazione più recenti.

```
$ sudo opsworks-agent-cli get_json configure

{
```

```
  "run_list": [
    "recipe[opsworks_cookbook_demo::configure]"
  ]
}
```

Il seguente esempio di output mostra le informazioni che AWS OpsWorks Stacks utilizza internamente per l'esecuzione di Chef eseguita per il timestamp specificato.

```
$ sudo opsworks-agent-cli get_json 2015-12-01T18:20:24 -i

{
  "aws_opsworks_agent": {
    "version": "4004-20151201152533",
    "valid_client_activities": [
      "reboot",
      "stop",
      "deploy",
      "grant_remote_access",
      "revoke_remote_access",
      "update_agent",
      "setup",
      "configure",
      "update_dependencies",
      "install_dependencies",
      "update_custom_cookbooks",
      "execute_recipes",
      "sync_remote_users"
    ],
    "command": {
      "type": "configure",
      "args": {
        "app_ids": [

        ]
      },
      "sent_at": "2015-12-01T18:19:23+00:00",
      "command_id": "5c2113f3-c6d5-40eb-bcfa-77da2885eeEX",
      "iam_user_arn": null,
      "instance_id": "cfdaa716-42fe-4e3b-9762-fef184ddd8EX"
    },
    "resources": {
      "apps": [
```

```
      ],
      "layers": [
        {
          "layer_id": "93f50d83-1e73-45c4-840a-0d4f07cda1EX",
          "name": "MyCookbooksDemoLayer",
          "packages": [

          ],
          "shortname": "cookbooks-demo",
          "type": "custom",
          "volume_configurations": [

          ]
        }
      ],
      "instances": [
        {
          "ami_id": "ami-d93622EX",
          "architecture": "x86_64",
          "auto_scaling_type": null,
          "availability_zone": "us-west-2a",
          "created_at": "2015-11-18T00:21:05+00:00",
          "ebs_optimized": false,
          "ec2_instance_id": "i-a480e960",
          "elastic_ip": null,
          "hostname": "cookbooks-demo1",
          "instance_id": "cfdaa716-42fe-4e3b-9762-fef184ddd8EX",
          "instance_type": "c3.large",
          "layer_ids": [
            "93f50d83-1e73-45c4-840a-0d4f07cda1EX"
          ],
          "os": "Amazon Linux 2015.09",
          "private_dns": "ip-192-0-2-0.us-west-2.compute.internal",
          "private_ip": "10.122.69.33",
          "public_dns": "ec2-203-0-113-0.us-west-2.compute.amazonaws.com",
          "public_ip": "192.0.2.0",
          "root_device_type": "ebs",
          "root_device_volume_id": "vol-f6f7e8EX",
          "ssh_host_dsa_key_fingerprint": "f2:...:15",
          "ssh_host_dsa_key_public": "ssh-dss AAAAB3Nz...a8vMbqA=",
          "ssh_host_rsa_key_fingerprint": "0a:...:96",
          "ssh_host_rsa_key_public": "ssh-rsa AAAAB3Nz...yhPanvo7",
          "status": "online",
          "subnet_id": null,
```

```
          "virtualization_type": "paravirtual",
          "infrastructure_class": "ec2",
          "ssh_host_dsa_key_private": "-----BEGIN DSA PRIVATE KEY-----
\nMIIDVwIB...g5OtgQ==\n-----END DSA PRIVATE KEY-----\n",
          "ssh_host_rsa_key_private": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIB...78kprtIw\n-----END RSA PRIVATE KEY-----\n"
        }
      ],
      "users": [

      ],
      "elastic_load_balancers": [

      ],
      "rds_db_instances": [

      ],
      "stack": {
        "arn": "arn:aws:opsworks:us-west-2:80398EXAMPLE:stack/040c3def-b2b4-4489-bb1b-
e08425886fEX/",
        "custom_cookbooks_source": {
          "type": "s3",
          "url": "https://s3.amazonaws.com/amzn-s3-demo-bucket/opsworks-cookbook-
demo.tar.gz",
          "username": "AKIAJUQN...WG644EXA",
          "password": "O5v+4Zz+...rcKbFTJu",
          "ssh_key": null,
          "revision": null
        },
        "name": "MyCookbooksDemoStack",
        "region": "us-west-2",
        "stack_id": "040c3def-b2b4-4489-bb1b-e08425886fEX",
        "use_custom_cookbooks": true,
        "vpc_id": null
      },
      "ecs_clusters": [

      ],
      "volumes": [

      ]
    },
    "chef": {
      "customer_recipes": [
```

```
        "opsworks_cookbook_demo::configure"
      ],
      "customer_json": "e30=\n",
      "customer_data_bags": "e30=\n"
    }
  }
}
```

instance_report

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Restituisce un report dell'istanza estesa.

```
sudo opsworks-agent-cli instance_report
```

Il seguente output di esempio proviene da un'istanza.

```
$ sudo opsworks-agent-cli instance_report

AWS OpsWorks Instance Agent State Report:

  Last activity was a "configure" on 2015-12-01 18:19:23 UTC
  Agent Status: The AWS OpsWorks agent is running as PID 30998
  Agent Version: 4004-20151201152533, up to date
  OpsWorks Stack: MyCookbooksDemoStack
  OpsWorks Layers: MyCookbooksDemoLayer
  OpsWorks Instance: cookbooks-demo1
  EC2 Instance ID: i-a480e9EX
  EC2 Instance Type: c3.large
  Architecture: x86_64
  Total Memory: 3.84 Gb
  CPU: 2x  Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
```

```
Location:

  EC2 Region: us-west-2
  EC2 Availability Zone: us-west-2a

Networking:

  Public IP: 192.0.2.0
  Private IP: 198.51.100.0
```

list_commands

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Elenca i tempi di ogni attività che è stata eseguita su questa istanza. Puoi utilizzare questi tempi per altri comandi della CLI dell'agente, al fine di specificare una determinata esecuzione.

```
sudo opsworks-agent-cli list_commands  [activity] [date]
```

Il seguente output di esempio proviene da un'istanza che ha eseguito le attività di configurazione, impostazione e aggiornamento dei libri di ricette personalizzati.

```
$ sudo opsworks-agent-cli list_commands

2015-11-24T21:00:28        update_custom_cookbooks
2015-12-01T18:19:09        setup
2015-12-01T18:20:24        configure
```

run_command

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Esegue un comando AWS OpsWorks Stacks, che è un file JSON contenente un run-list di Chef che contiene le informazioni necessarie per eseguire un'attività AWS OpsWorks Stacks (installazione, configurazione, distribuzione e così via). Il comando `run_command` genera una voce di log che puoi visualizzare mediante l'esecuzione di `show_log`. Questa opzione è destinata esclusivamente allo sviluppo, quindi AWS OpsWorks Stacks non tiene traccia delle modifiche.

```
sudo opsworks-agent-cli run_command [activity] [date] [/path/to/valid/json.file]
```

Per impostazione predefinita, `run_command` esegue il comando AWS OpsWorks Stacks più recente. Utilizza le seguenti opzioni per specificare un determinato comando.

activity

Esegue un comando AWS OpsWorks Stacks specificato:`setup`,`configure`,`deploy`,, `undeploy` `startstop`, o. `restart`

data

Esegui il OpsWorks comando AWS eseguito nel timestamp specificato. Per ottenere un elenco dei Time stamp validi, esegui `list_commands`.

file

Esegui il file JSON del comando specificato. Per ottenere un percorso del file del comando, esegui `get_json`.

Il seguente output di esempio proviene da un'istanza ed esegue il comando di configurazione.

```
$ sudo opsworks-agent-cli run_command configure
```

```
[2015-12-02 16:52:53]  INFO [opsworks-agent(21970)]: About to re-run 'configure' from
 2015-12-01T18:20:24
...
[2015-12-02 16:53:02]  INFO [opsworks-agent(21970)]: Finished Chef run with exitcode 0
```

show_log

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Restituisce un file di log del comando.

```
sudo opsworks-agent-cli show_log [activity] [date]
```

Per impostazione predefinita, `show_log` accoda il file di log più recente. Utilizza le seguenti opzioni per specificare un determinato comando.

activity

Mostra il file di log dell'attività specificata.

data

Mostra il file di log per l'attività eseguita per un Time stamp specificato. Per ottenere un elenco dei Time stamp validi, esegui `list_commands`.

Il seguente esempio di output mostra il log più recente.

```
$ sudo opsworks-agent-cli show_log

[2015-12-02T16:52:59+00:00] INFO: Storing updated cookbooks/opsworks_cookbook_demo/
opsworks-cookbook-demo.tar.gz in the cache.
...
```

```
[2015-12-02T16:52:59+00:00] INFO: Report handlers complete
```

stack_state

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Visualizza le informazioni che AWS OpsWorks Stacks utilizza internamente per l'ultima esecuzione di Chef.

```
opsworks-agent-cli stack_state
```

> **Note**
>
> Per le istanze Linux di Chef 12, l'esecuzione di questo comando restituirà delle informazioni valide, ad esempio la configurazione dello stack e gli attributi di configurazione dell'istanza. Tuttavia, per ottenere informazioni più complete, fai riferimento ai data bag Chef che AWS OpsWorks Stacks crea sull'istanza. Per ulteriori informazioni, consulta la AWS OpsWorks Riferimento a Stacks Data Bag.

Il seguente output di esempio proviene da un'istanza.

```
$ sudo opsworks-agent-cli stack_state

{
  "last_command": {
    "sent_at": "2015-12-01T18:19:23+00:00",
    "activity": "configure"
  },
  "instance": {
    "ami_id": "ami-d93622EX",
    "architecture": "x86_64",
```

```
    "auto_scaling_type": null,
    "availability_zone": "us-west-2a",
    "created_at": "2015-11-18T00:21:05+00:00",
    "ebs_optimized": false,
    "ec2_instance_id": "i-a480e9EX",
    "elastic_ip": null,
    "hostname": "cookbooks-demo1",
    "instance_id": "cfdaa716-42fe-4e3b-9762-fef184ddd8EX",
    "instance_type": "c3.large",
    "layer_ids": [
      "93f50d83-1e73-45c4-840a-0d4f07cda1EX"
    ],
    "os": "Amazon Linux 2015.09",
    "private_dns": "ip-192-0-2-0.us-west-2.compute.internal",
    "private_ip": "10.122.69.33",
    "public_dns": "ec2-203-0-113-0.us-west-2.compute.amazonaws.com",
    "public_ip": "192.0.2.0",
    "root_device_type": "ebs",
    "root_device_volume_id": "vol-f6f7e8EX",
    "ssh_host_dsa_key_fingerprint": "f2:...:15",
    "ssh_host_dsa_key_public": "ssh-dss AAAAB3Nz...a8vMbqA=",
    "ssh_host_rsa_key_fingerprint": "0a:...:96",
    "ssh_host_rsa_key_public": "ssh-rsa AAAAB3Nz...yhPanvo7",
    "status": "online",
    "subnet_id": null,
    "virtualization_type": "paravirtual",
    "infrastructure_class": "ec2",
    "ssh_host_dsa_key_private": "-----BEGIN DSA PRIVATE KEY-----\nMIIDVwIB...g5OtgQ==
\n-----END DSA PRIVATE KEY-----\n",
    "ssh_host_rsa_key_private": "-----BEGIN RSA PRIVATE KEY-----\nMIIEowIB...78kprtIw
\n-----END RSA PRIVATE KEY-----\n"
  },
  "layers": [
    {
      "layer_id": "93f50d83-1e73-45c4-840a-0d4f07cda1EX",
      "name": "MyCookbooksDemoLayer",
      "packages": [

      ],
      "shortname": "cookbooks-demo",
      "type": "custom",
      "volume_configurations": [

      ]
```

```
    }
  ],
  "applications": null,
  "stack": {
    "arn": "arn:aws:opsworks:us-west-2:80398EXAMPLE:stack/040c3def-b2b4-4489-bb1b-
e08425886fEX/",
    "custom_cookbooks_source": {
      "type": "s3",
      "url": "https://s3.amazonaws.com/amzn-s3-demo-bucket/opsworks-cookbook-
demo.tar.gz",
      "username": "AKIAJUQN...WG644EXA",
      "password": "O5v+4Zz+...rcKbFTJu",
      "ssh_key": null,
      "revision": null
    },
    "name": "MyCookbooksDemoStack",
    "region": "us-west-2",
    "stack_id": "040c3def-b2b4-4489-bb1b-e08425886fEX",
    "use_custom_cookbooks": true,
    "vpc_id": null
  },
  "agent": {
    "valid_activities": [
      "reboot",
      "stop",
      "deploy",
      "grant_remote_access",
      "revoke_remote_access",
      "update_agent",
      "setup",
      "configure",
      "update_dependencies",
      "install_dependencies",
      "update_custom_cookbooks",
      "execute_recipes",
      "sync_remote_users"
    ]
  }
}
```

AWS OpsWorks Riferimento a Stacks Data Bag

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

AWS OpsWorks Stacks espone un'ampia varietà di impostazioni alle ricette come contenuto del data bag di Chef. In questo documento di riferimento viene specificato il contenuto dei contenitori di dati.

Un contenitore di dati è un concetto Chef. Si tratta di una variabile globale che viene archiviata sotto forma di dati JSON su un'istanza. I dati JSON sono accessibili da Chef. Ad esempio, un data bag può memorizzare variabili globali come l'URL di origine di un'app, il nome host dell'istanza e l'identificatore VPC dello stack associato. AWS OpsWorks Stacks archivia i propri data bag nelle istanze di ogni stack. Nelle istanze Linux, AWS OpsWorks Stacks archivia i sacchetti di dati nella directory. `/var/chef/runs/`*`run-ID`*`/data_bags` Sulle istanze Windows, archivia i contenitori di dati nella directory *`drive`*`:\chef\runs\`*`run-id`*`\data_bags`. In entrambi i casi, *`run-ID`* è un ID univoco che AWS OpsWorks Stacks assegna a ogni Chef eseguito su un'istanza. Queste directory includono un set di contenitori di dati (sottodirectory). Ogni contenitore di dati include zero o più elementi di contenitori di dati, ovvero file in formato JSON che includono i gruppi di contenuti dei contenitori di dati.

> **Note**
>
> AWS OpsWorks Stacks non supporta sacchetti di dati crittografati. Per archiviare i dati riservati in forma crittografata, ad esempio le password o i certificati, ti consigliamo di archiviarli in un bucket S3 privato. Potrai quindi creare una ricetta personalizzata che utilizza l'SDK Amazon per Ruby per recuperare i dati. Per vedere un esempio, consulta Utilizzo dell'SDK for Ruby.

Il contenuto dei contenitori di dati può includere gli elementi seguenti:

- Contenuto stringa che segue una sintassi Ruby standard e può utilizzare virgolette singole o doppie, sebbene le stringhe contenenti determinati caratteri speciali debbano avere virgolette doppie. Per ulteriori informazioni, consulta il sito della documentazione di Ruby.
- Contenuto booleano, ovvero `true` o `false` (senza virgolette).
- Contenuto numerico, che può corrispondere a numeri interi o decimali, ad esempio `4` o `2.5` (senza virgolette).
- Contenuto elenco, nella forma di valori separati da virgola e racchiusi tra parentesi quadre (senza virgolette), ad esempio `[ '80', '443' ]`
- Gli oggetti JSON, che includono contenuto aggiuntivo dei contenitori di dati, ad esempio `"my-app": {"elastic_ip": null,...}`.

Le ricette Chef possono accedere a contenitori di dati, elementi di contenitori di dati e contenuto di contenitori di dati tramite la ricerca Chef oppure direttamente. Di seguito viene descritto come utilizzare entrambi gli approcci di accesso (sebbene la ricerca Chef sia da preferire).

Per accedere a un data bag tramite Chef search, usa il metodo di ricerca, specificando l'indice di ricerca desiderato. AWS OpsWorks Stacks fornisce i seguenti indici di ricerca:

- aws_opsworks_app, che rappresenta un gruppo di app distribuite per uno stack.
- aws_opsworks_command, che rappresenta un gruppo di comandi eseguiti su uno stack.
- aws_opsworks_ecs_cluster, che rappresenta un set di istanze di cluster Amazon Elastic Container Service (Amazon ECS) per uno stack.
- aws_opsworks_elastic_load_balancer, che rappresenta un set di sistemi di bilanciamento del carico Elastic Load Balancing per uno stack.
- aws_opsworks_instance, che rappresenta un gruppo di istanze per uno stack.
- aws_opsworks_layer, che rappresenta un gruppo di livelli per uno stack.
- aws_opsworks_rds_db_instance, che rappresenta un set di istanze Amazon Relational Database Service (Amazon RDS) per uno stack.
- aws_opsworks_stack, che rappresenta uno stack.
- aws_opsworks_user, che rappresenta un gruppo di utenti per uno stack.

Una volta noto il nome dell'indice di ricerca, puoi accedere al contenuto del contenitore di dati per tale indice di ricerca. Ad esempio, il seguente codice ricetta utilizza l'indice di ricerca

`aws_opsworks_app` per ottenere il contenuto del primo elemento del contenitore di dati (il primo file JSON) nel contenitore di dati `aws_opsworks_app` (la directory `aws_opsworks_app`). Il codice, quindi, scrive due messaggi nel log di Chef, uno con il contenuto del contenitore di dati del nome breve dell'app (una stringa nel file JSON) e l'altro con il contenuto del contenitore di dati dell'URL di origine dell'app (un'altra stringa nel file JSON):

```
app = search("aws_opsworks_app").first
Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}' **********")
```

Dove `['shortname']` e `['app_source']['url']` specificano il contenuto del contenitore di dati seguente nel file JSON corrispondente:

```
{
  ...
  "shortname": "mylinuxdemoapp",
  ...
  "app_source": {
    ...
    "url": "https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-
nodejs.tar.gz",
  },
  ...
}
```

Per ottenere un elenco del contenuto dei contenitori di dati che puoi cercare, consulta gli argomenti di riferimento in questa sezione.

Puoi anche eseguire l'iterazione di un set di elementi di un contenitore di dati. Ad esempio, il codice ricetta seguente è simile all'esempio precedente in quanto esegue l'iterazione di ogni elemento del contenitore di dati quando sono presenti più elementi:

```
search("aws_opsworks_app").each do |app|
  Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
  Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}'
 **********")
end
```

Se sei a conoscenza dell'esistenza di contenuto nel contenitore di dati specifico, puoi trovare l'elemento del contenitore di dati corrispondente con la sintassi seguente:

```
search("search_index", "key:value").first
```

Ad esempio, il codice ricetta seguente utilizza l'indice di ricerca `aws_opsworks_app` per trovare l'elemento del contenitore di dati che contiene il nome breve `mylinuxdemoapp` dell'app. Quindi, utilizza i contenuti dell'elemento del contenitore di dati per scrivere un messaggio nel log di Chef con l'URL di origine e il nome breve dell'app corrispondente:

```
app = search("aws_opsworks_app", "shortname:mylinuxdemoapp").first
Chef::Log.info("********** For the app with the short name '#{app['shortname']}', the
 app's URL is '#{app['app_source']['url']}' **********")
```

Solo per l'indice di ricerca `aws_opsworks_instance`, puoi specificare `self:true` per rappresentare l'istanza su cui viene eseguita la ricetta. Il seguente codice di ricetta utilizza il contenuto dell'elemento del data bag corrispondente per scrivere un messaggio nel registro Chef con l'ID e il sistema operativo generati da Stacks dell'istanza corrispondente: AWS OpsWorks

```
instance = search("aws_opsworks_instance", "self:true").first
Chef::Log.info("********** For instance '#{instance['instance_id']}', the instance's
 operating system is '#{instance['os']}' **********")
```

Puoi accedere direttamente ai contenitori di dati, agli elementi dei contenitori di dati e al contenuto dei contenitori di dati senza dover necessariamente utilizzare la ricerca Chef. A tale scopo, utilizza i metodi data_bag e data_bag_item per accedere, rispettivamente, ai contenitori di dati e agli elementi dei contenitori di dati. Ad esempio, il seguente codice ricetta esegue la stessa operazione degli esempi precedenti, ad eccezione del fatto che accede direttamente a un singolo elemento del contenitore di dati, quindi a più elementi del contenitore di dati, se ve ne sono più di uno:

```
# Syntax: data_bag_item("the data bag name", "the file name in the data bag without the
 file extension")
app = data_bag_item("aws_opsworks_app", "mylinuxdemoapp")
Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}' **********")

data_bag("aws_opsworks_app").each do |data_bag_item|
  app = data_bag_item("aws_opsworks_app", data_bag_item)
  Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
  Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}'
 **********")
end
```

Di questi due approcci, ti consigliamo di usare la ricerca Chef. Tutti gli esempi correlati in questa guida dimostrano questo approccio.

Argomenti

- Contenitore di dati di app (aws_opsworks_app)
- Contenitore di dati di comando (aws_opsworks_command)
- Borsa dati del cluster Amazon ECS (aws_opsworks_ecs_cluster)
- Borsa dati Elastic Load Balancing (aws_opsworks_elastic_load_balancer)
- Contenitore di dati dell'istanza (aws_opsworks_instance)
- Contenitori di dati di livello (aws_opsworks_layer)
- Borsa dati Amazon RDS (aws_opsworks_rds_db_instance)
- Contenitore di dati dello stack (aws_opsworks_stack)
- Contenitore di dati utente (aws_opsworks_user)

Contenitore di dati di app (aws_opsworks_app)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Per un evento Deploy o un comando dello stack Execute Recipes (Esegui ricette), rappresenta le impostazioni di un'app.

L'esempio seguente mostra come utilizzare Chef search per cercare in un singolo elemento del data bag e poi in più elementi del data bag per scrivere messaggi nel registro Chef con i nomi brevi e la fonte delle app: URLs

```
app = search("aws_opsworks_app").first
Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}' **********")
```

```
search("aws_opsworks_app").each do |app|
  Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
  Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}'
 **********")
end
```

app_id	app_source	data_sources
distribuisci	attributes	domains
enable_ssl	ambiente	nome
shortname	ssl_configuration	tipo

app_id

L'ID dell'app (stringa). Un GUID che identifica l'app.

app_source

Un set di contenuti che specifica le informazioni utilizzate da AWS OpsWorks Stacks per distribuire l'app dal suo repository di controllo del codice sorgente. Il contenuto varia in base al tipo di repository.

password

La password per i repository privati e `"null"` per i repository pubblici (stringa). Per bucket S3 privati, questo contenuto è impostato sulla chiave segreta.

revisione

Se nel repository sono presenti più rami, il contenuto specifica il ramo o la versione dell'app, ad esempio `"version1"` (stringa). Altrimenti il valore è impostato su `"null"`.

ssh_key

Una chiave SSH di distribuzione per l'accesso ai repository Git privati e `"null"` per i repository pubblici (stringa).

tipo

Il percorso di origine dell'app (stringa). I valori validi includono:

- "archive"
- "git"
- "other"
- "s3"

url

Dove si trova l'origine dell'app (stringa).

Utente

Il nome utente per i repository privati e "null" per i repository pubblici (stringa). Per bucket S3 privati, il contenuto è impostato sulla chiave di accesso.

attributes

Un set di contenuti che descrive la struttura della directory e il contenuto dell'app.

document_root

La directory radice della struttura del documento. Definisce il percorso alla radice del documento, oppure il percorso della home page dell'app, ad esempio home_html, relativo alla tua directory di distribuzione. Se non si specifica questo attributo, l'impostazione predefinita di document_root sarà public. Il valore di document_root può iniziare solo con i caratteri a-z, A-Z, 0-9, _ (carattere di sottolineatura) o - (trattino).

data_sources

Le informazioni richieste per la connessione al database dell'app. Se all'app è collegato un livello di database, AWS OpsWorks Stacks assegna automaticamente i valori appropriati a questo contenuto.

Il valore di data_sources è una matrice e le matrici sono accessibili mediante uno scostamento integrale, non mediante una chiave. Ad esempio, per accedere alla prima origine dati dell'app, utilizza app[:data_sources][0][:type].

database_name

Il nome del database, che di norma è il nome breve dell'app (stringa).

tipo

Il tipo dell'istanza di database, generalmente "RdsDbInstance" (stringa).

arn

L'Amazon Resource Name (ARN) dell'istanza di database (stringa).

distribuisci

Se l'app deve essere distribuita (booleano). `true` per le app che devono essere distribuite in un evento del ciclo di vita Deploy. In un evento del ciclo di vita Setup, questo contenuto sarà `true` per tutte le app. Per determinare quali app devono essere distribuite su un'istanza, verifica i livelli a cui l'istanza appartiene.

domains

Un elenco dei domini dell'app (elenco di stringhe).

enable_ssl

Se il supporto SSL è abilitato (booleano).

ambiente

Una raccolta di variabili di ambiente specificate dall'utente, definite per l'app. Per ulteriori informazioni su come definire le variabili di ambiente di un'app, consulta Aggiunta di app. Ogni nome di contenuto è impostato su un nome di variabile di ambiente e il valore corrispondente è impostato sul valore della variabile.

nome

Il nome dell'app, utilizzato ai fini della visualizzazione (stringa).

shortname

Il nome breve dell'app, generato da AWS OpsWorks Stacks a partire dal nome (stringa). Il nome breve viene utilizzato internamente e dalle ricette. Viene usato come nome della directory in cui sono installati i file dell'app.

ssl_configuration

certificate

Se hai abilitato il supporto SSL, il certificato SSL dell'app, altrimenti `"null"` (stringa).

chain

Se SSL è abilitato, il contenuto per specificare una chiave di autorità di certificazione intermedia o l'autenticazione client (stringa).

private_key

Se hai abilitato il supporto SSL, la chiave privata SSL dell'app, altrimenti `"null"` (stringa).

tipo

Il tipo dell'app, che è sempre impostato su `"other"` per gli stack Chef 12 Linux e Chef 12.2 Windows (stringa).

Contenitore di dati di comando (aws_opsworks_command)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Rappresenta le impostazioni per un comando che AWS OpsWorks Stacks esegue su una o più istanze.

L'esempio seguente mostra come usare la ricerca Chef per cercare in un singolo elemento del contenitore di dati, quindi in più elementi del contenitore di dati per scrivere messaggi nel log di Chef con i tipi di comandi e il momento in cui sono stati inviati:

```
command = search("aws_opsworks_command").first
Chef::Log.info("********** The command's type is '#{command['type']}' **********")
Chef::Log.info("********** The command was sent at '#{command['sent_at']}' **********")

search("aws_opsworks_command").each do |command|
  Chef::Log.info("********** The command's type is '#{command['type']}' **********")
  Chef::Log.info("********** The command was sent at '#{command['sent_at']}'
 **********")
end
```

args	command_id	iam_user_arn

instance_id	sent_at	tipo

args

Argomenti del comando (stringa).

command_id

L'identificatore univoco casuale del comando, assegnato da AWS OpsWorks Stacks (string).

iam_user_arn

Se il comando viene creato dal cliente, l'Amazon Resource Name (ARN) dell'utente che ha creato il comando (stringa).

instance_id

L'identificatore dell'istanza su cui il comando è stato eseguito (stringa).

sent_at

Il timestamp di quando AWS OpsWorks Stacks ha eseguito il comando (stringa).

tipo

Il tipo del comando (stringa). I valori validi includono:

- `"configure"`
- `"deploy"`
- `"deregister"`
- `"execute_recipes"`
- `"grant_remote_access"`
- `"install_dependencies"`
- `"restart"`
- `"revoke_remote_access"`
- `"rollback"`
- `"setup"`
- `"shutdown"`

- "start"
- "stop"
- "sync_remote_users"
- "undeploy"
- "update_agent"
- "update_custom_cookbooks"
- "update_dependencies"

Borsa dati del cluster Amazon ECS (aws_opsworks_ecs_cluster)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Rappresenta le impostazioni di un cluster Amazon ECS.

L'esempio seguente mostra come utilizzare Chef search per cercare un singolo elemento del data bag e poi più elementi del data bag per scrivere messaggi nel registro Chef con i nomi dei cluster Amazon ECS e Amazon Resource Names ()ARNs:

```
ecs_cluster = search("aws_opsworks_ecs_cluster").first
Chef::Log.info("********** The ECS cluster's name is
 '#{ecs_cluster['ecs_cluster_name']}' **********")
Chef::Log.info("********** The ECS cluster's ARN is '#{ecs_cluster['ecs_cluster_arn']}'
 **********")

search("aws_opsworks_ecs_cluster").each do |ecs_cluster|
  Chef::Log.info("********** The ECS cluster's name is
 '#{ecs_cluster['ecs_cluster_name']}' **********")
  Chef::Log.info("********** The ECS cluster's ARN is
 '#{ecs_cluster['ecs_cluster_arn']}' **********")
end
```

ecs_cluster_arn	ecs_cluster_name

ecs_cluster_arn

L'Amazon Resource Name (ARN) del cluster (stringa).

ecs_cluster_name

Il nome del cluster (stringa).

Borsa dati Elastic Load Balancing (aws_opsworks_elastic_load_balancer)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Rappresenta le impostazioni di un sistema di bilanciamento del carico Elastic Load Balancing.

L'esempio seguente mostra come utilizzare Chef search per cercare un singolo elemento del data bag e poi più elementi del data bag per scrivere messaggi nel registro Chef con i nomi e i nomi DNS dei sistemi di bilanciamento del carico Elastic Load Balancing:

```
elastic_load_balancer = search("aws_opsworks_elastic_load_balancer").first
Chef::Log.info("********** The ELB's name is
 '#{elastic_load_balancer['elastic_load_balancer_name']}' **********")
Chef::Log.info("********** The ELB's DNS name is '#{elastic_load_balancer['dns_name']}'
 **********")

search("aws_opsworks_elastic_load_balancer").each do |elastic_load_balancer|
  Chef::Log.info("********** The ELB's name is
 '#{elastic_load_balancer['elastic_load_balancer_name']}' **********")
  Chef::Log.info("********** The ELB's DNS name is
 '#{elastic_load_balancer['dns_name']}' **********")
end
```

elastic_load_balancer_name	dns_name	layer_id

elastic_load_balancer_name

Il nome del sistema di bilanciamento del carico (stringa).

dns_name

Il nome DNS del sistema di bilanciamento del carico (stringa).

layer_id

L'ID AWS OpsWorks Stacks del layer a cui è assegnato il load balancer (stringa).

Contenitore di dati dell'istanza (aws_opsworks_instance)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Rappresenta le impostazioni di un'istanza.

L'esempio seguente mostra come utilizzare Chef search per cercare in un singolo elemento del data bag e poi in più elementi del data bag per scrivere messaggi nel registro Chef con i nomi host delle istanze e: IDs

```
instance = search("aws_opsworks_instance").first
Chef::Log.info("********** The instance's hostname is '#{instance['hostname']}'
 **********")
Chef::Log.info("********** The instance's ID is '#{instance['instance_id']}'
 **********")

search("aws_opsworks_instance").each do |instance|
  Chef::Log.info("********** The instance's hostname is '#{instance['hostname']}'
 **********")
```

```
  Chef::Log.info("********** The instance's ID is '#{instance['instance_id']}'
 **********")
end
```

L'esempio seguente mostra diversi modi di utilizzare Chef search per cercare tra più elementi del data bag per trovare l'elemento del data bag che contiene l'ID di EC2 istanza Amazon specificato. L'esempio utilizza, quindi, i contenuti dell'elemento del contenitore di dati per scrivere un messaggio nel log di Chef con l'indirizzo IP pubblico dell'istanza corrispondente:

```
instance = search("aws_opsworks_instance", "ec2_instance_id:i-12345678").first
Chef::Log.info("********** For instance '#{instance['ec2_instance_id']}', the
 instance's public IP address is '#{instance['public_ip']}' **********")

search("aws_opsworks_instance").each do |instance|
  if instance['ec2_instance_id'] == 'i-12345678'
    Chef::Log.info("********** For instance '#{instance['ec2_instance_id']}', the
 instance's public IP address is '#{instance['public_ip']}' **********")
  end
end
```

L'esempio seguente mostra come usare la ricerca Chef con `self:true` per trovare l'elemento del contenitore di dati contenente informazioni correlate all'istanza su cui viene eseguita la ricetta. L'esempio utilizza quindi il contenuto dell'elemento del data bag per scrivere un messaggio nel registro Chef con l'ID AWS OpsWorks generato da Stacks dell'istanza corrispondente e l'indirizzo IP pubblico dell'istanza:

```
instance = search("aws_opsworks_instance", "self:true").first
Chef::Log.info("********** For instance '#{instance['instance_id']}', the instance's
 public IP address is '#{instance['public_ip']}' **********")
```

ami_id	architecture	auto_scaling_type
availability_zone	created_at	ebs_optimized
ec2_instance_id	elastic_ip	hostname
instance_id	instance_type	layer_ids
so	private_dns	private_ip

public_dns	public_ip	root_device_type
root_device_volume_id	self	ssh_host_dsa_key_fingerprint
ssh_host_dsa_key_private	ssh_host_dsa_key_public	ssh_host_rsa_key_fingerprint
ssh_host_rsa_key_private	ssh_host_rsa_key_public	status
subnet_id	virtualization_type	

ami_id

L'ID dell'AMI (Amazon Machine Image) dell'istanza (stringa).

architecture

L'architettura dell'istanza, che è sempre impostata su `"x86_64"` (stringa).

auto_scaling_type

Il tipo di dimensionamento dell'istanza: `null`, `timer` o `load` (stringa).

availability_zone

La zona di disponibilità (AZ) dell'istanza, ad esempio `"us-west-2a"` (stringa).

created_at

La data in cui l'istanza è stata creata, nel formato UTC "*yyyy*-*mm*-*ddd*T*hh*:*mm*:*ss*+*hh*:*mm*" (stringa). Ad esempio, `"2013-10-01T08:35:22+00:00"` corrisponde a 8:35:22 del 10 ott. 2013, senza differenza di fuso orario. Per ulteriori informazioni, consulta il protocollo ISO 8601.

ebs_optimized

Se l'istanza è ottimizzata per EBS (booleano).

ec2_instance_id

L'ID dell' EC2 istanza (stringa).

elastic_ip

L'indirizzo IP elastico, impostato su `"null"` se l'istanza non dispone di un indirizzo IP elastico (stringa).

hostname

Il nome host, ad esempio `"demo1"` (stringa).

instance_id

L'ID dell'istanza, che è un GUID AWS OpsWorks generato da Stacks che identifica in modo univoco l'istanza (stringa).

instance_type

Il tipo di istanza, ad esempio `"c1.medium"` (stringa).

layer_ids

Un elenco dei livelli dell'istanza, identificati dalla loro unicità; ad esempio,. IDs `307ut64c-c7e4-40cc-52f0-67d5k1f9992c`

so

Il sistema operativo dell'istanza (stringa). I valori validi includono:

- `"Amazon Linux 2"`
- `"Amazon Linux 2018.03"`
- `"Amazon Linux 2017.09"`
- `"Amazon Linux 2017.03"`
- `"Amazon Linux 2016.09"`
- `"Custom"`
- `"Microsoft Windows Server 2022 Base"`
- `"Microsoft Windows Server 2022 with SQL Server Express"`
- `"Microsoft Windows Server 2022 with SQL Server Standard"`
- `"Microsoft Windows Server 2022 with SQL Server Web"`
- `"Microsoft Windows Server 2019 Base"`
- `"Microsoft Windows Server 2019 with SQL Server Express"`
- `"Microsoft Windows Server 2019 with SQL Server Standard"`
- `"Microsoft Windows Server 2019 with SQL Server Web"`
- `"CentOS 7"`
- `"Red Hat Enterprise Linux 7"`
- `"Ubuntu 20.04 LTS"`

- "Ubuntu 18.04 LTS"
- "Ubuntu 16.04 LTS"
- "Ubuntu 14.04 LTS"

private_dns

Il nome DNS privato (stringa).

private_ip

L'indirizzo IP privato (stringa).

public_dns

Il nome DNS pubblico (stringa).

public_ip

L'indirizzo IP pubblico (stringa).

root_device_type

Il tipo di dispositivo radice (stringa). I valori validi includono:

- "ebs
- "instance-store"

root_device_volume_id

L'ID di volume del dispositivo radice (stringa).

self

true se questo elemento di contenitore di dati include informazioni sull'istanza su cui la ricetta viene eseguita, altrimenti false (booleano). Questo valore è disponibile solo per le ricette, non tramite l'API AWS OpsWorks Stacks.

ssh_host_dsa_key_fingerprint

Una sequenza di byte più corta che identifica la chiave pubblica DSA più lunga (stringa).

ssh_host_dsa_key_private

La chiave privata generata da DSA per l'autenticazione SSH con l'istanza (stringa).

ssh_host_dsa_key_public

La chiave pubblica generata da DSA per l'autenticazione SSH con l'istanza (stringa).

ssh_host_rsa_key_fingerprint

Una sequenza di byte più corta che identifica la chiave pubblica RSA più lunga (stringa).

ssh_host_rsa_key_private

La chiave privata generata da RSA per l'autenticazione SSH con l'istanza (stringa).

ssh_host_rsa_key_public

La chiave pubblica generata da RSA per l'autenticazione SSH con l'istanza (stringa).

status

Lo stato dell'istanza (stringa). I valori validi includono:

- `"requested"`
- `"booting"`
- `"running_setup"`
- `"online"`
- `"setup_failed"`
- `"start_failed"`
- `"terminating"`
- `"terminated"`
- `"stopped"`
- `"connection_lost"`

subnet_id

L'ID sottorete dell'istanza (stringa).

virtualization_type

Il tipo di virtualizzazione dell'istanza (stringa).

Contenitori di dati di livello (aws_opsworks_layer)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti

di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Rappresenta le impostazioni di un livello.

L'esempio seguente mostra come usare la ricerca Chef per cercare in un singolo elemento del contenitore di dati, quindi in più elementi del contenitore di dati per scrivere messaggi nel log di Chef con i nomi e i nomi brevi dei livelli:

```
layer = search("aws_opsworks_layer").first
Chef::Log.info("********** The layer's name is '#{layer['name']}' **********")
Chef::Log.info("********** The layer's shortname is '#{layer['shortname']}'
 **********")

search("aws_opsworks_layer").each do |layer|
  Chef::Log.info("********** The layer's name is '#{layer['name']}' **********")
  Chef::Log.info("********** The layer's shortname is '#{layer['shortname']}'
 **********")
end
```

ecs_cluster_arn	layer_id	nome
packages	shortname	tipo
volume_configurations		

ecs_cluster_arn

Se al layer è assegnato un cluster Amazon ECS, Amazon Resource Name (ARN) (stringa) del cluster Amazon ECS.

encrypted

`true` se il volume EBS è crittografato, altrimenti `false` (booleano).

layer_id

L'ID del livello, che è un GUID generato da AWS OpsWorks Stacks e che identifica in modo univoco il livello (stringa).

nome

Il nome del livello, che viene utilizzato per rappresentare il livello nella console (stringa). Può essere definito dall'utente e non deve necessariamente essere univoco.

packages

Un elenco di pacchetti da installare (elenco di stringhe).

shortname

Il nome breve del livello, che è definito dall'utente (stringa).

tipo

Il tipo del livello, che è sempre impostato su `"custom"` per Chef 12 Linux e Chef 12.2 Windows (stringa).

volume_configurations

Un elenco di configurazioni di volumi Amazon EBS.

iops

Il numero di operazioni I/O per secondo che il volume può supportare.

mount_point

La directory del punto di montaggio del volume.

number_of_disks

Il numero di dischi nel volume.

raid_level

Il livello di configurazione RAID del volume.

formato

La dimensione del volume in GiB.

volume_type

Il tipo di volume: di uso generale, magnetico, Provisioned IOPS, Throughput-Optimized HDD o Cold HDD.

Borsa dati Amazon RDS (aws_opsworks_rds_db_instance)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Un set di contenuti del data bag che specifica la configurazione di un'istanza Amazon Relational Database Service (Amazon RDS) come segue:

address	db_instance_identifier	db_password
db_user	engine	rds_db_instance_arn
Regione		

L'esempio seguente mostra come utilizzare Chef search per cercare un singolo elemento del data bag e poi più elementi del data bag per scrivere messaggi nel registro Chef con gli indirizzi delle istanze Amazon RDS e i tipi di motore di database:

```
rds_db_instance = search("aws_opsworks_rds_db_instance").first
Chef::Log.info("********** The RDS instance's address is
 '#{rds_db_instance['address']}' **********")
Chef::Log.info("********** The RDS instance's database engine type is
 '#{rds_db_instance['engine']}' **********")

search("aws_opsworks_rds_db_instance").each do |rds_db_instance|
  Chef::Log.info("********** The RDS instance's address is
 '#{rds_db_instance['address']}' **********")
  Chef::Log.info("********** The RDS instance's database engine type is
 '#{rds_db_instance['engine']}' **********")
end
```

address

Il nome DNS dell'istanza.

port

La porta dell'istanza.

db_instance_identifier

L'ID dell'istanza.

db_password

La password master dell'istanza.

db_user

Il nome utente master dell'istanza.

engine

Il motore di database dell'istanza, ad esempio `mysql`.

rds_db_instance_arn

L'Amazon Resource Name (ARN) dell'istanza.

Regione

La regione AWS dell'istanza, ad esempio `us-west-2`.

Contenitore di dati dello stack (aws_opsworks_stack)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Rappresenta le impostazioni di uno stack.

L'esempio seguente mostra come usare la ricerca Chef per scrivere messaggi nel log di Chef con il nome dello stack e l'URL di origine del libro di ricette:

```
stack = search("aws_opsworks_stack").first
Chef::Log.info("********** The stack's name is '#{stack['name']}' **********")
Chef::Log.info("********** The stack's cookbook URL is
 '#{stack['custom_cookbooks_source']['url']}' **********")
```

arn	custom_cookbooks_source	nome
Regione	stack_id	use_custom_cookbooks
vpc_id		

arn

L'Amazon Resource Name (ARN) dello stack (stringa).

custom_cookbooks_source

Una serie di contenuti che specifica il repository di origine del libro di ricette personalizzato.

tipo

Il tipo di repository (stringa). I valori validi includono:

- `"archive"`
- `"git"`
- `"s3"`

url

L'URL del repository, come `"git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git"` (stringa).

username

Il nome utente per i repository privati e `null` per i repository pubblici (stringa). Per i bucket privati di Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3), il contenuto è impostato sulla chiave di accesso.

password

La password per i repository privati e `null` per i repository pubblici (stringa). Per bucket S3 privati, questo contenuto è impostato sulla chiave segreta.

ssh_key

Una chiave SSH di distribuzione per l'accesso ai repository Git privati e `null` per i repository pubblici (stringa).

revisione

Se nel repository sono presenti più rami, il contenuto specifica il ramo o la versione dell'app, ad esempio `"version1"` (stringa). In alternativa, è impostato su `null`.

nome

Il nome dello stack (stringa).

Regione

La regione AWS dello stack (stringa).

stack_id

Un GUID che identifica lo stack (stringa).

use_custom_cookbooks

Indica se i libri di ricette personalizzati sono abilitati (booleano).

vpc_id

Se lo stack è in esecuzione in un VPC, l'ID VPC (stringa).

Contenitore di dati utente (aws_opsworks_user)

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Rappresenta le impostazioni di un utente.

L'esempio seguente mostra come utilizzare Chef search per cercare un singolo elemento del data bag e poi più elementi del data bag per scrivere messaggi nel registro Chef con i nomi utente degli utenti e Amazon Resource Names (ARNs):

```
user = search("aws_opsworks_user").first
Chef::Log.info("********** The user's user name is '#{user['username']}' **********")
Chef::Log.info("********** The user's user ARN is '#{user['iam_user_arn']}'
 **********")

# Or...

search("aws_opsworks_user").each do |user|
  Chef::Log.info("********** The user's user name is '#{user['username']}' **********")
  Chef::Log.info("********** The user's user ARN is '#{user['iam_user_arn']}'
 **********")
end
```

administrator_privileges	iam_user_arn	remote_access
ssh_public_key	unix_user_id	username

administrator_privileges

Indica se l'utente dispone di privilegi di amministratore (booleano).

iam_user_arn

L'Amazon Resource Name (ARN) dell'utente (stringa).

remote_access

Se l'utente può utilizzare RDP per accedere all'istanza (booleano).

ssh_public_key

La chiave pubblica dell'utente, fornita tramite la console AWS OpsWorks Stacks o l'API (stringa).

unix_user_id

L'ID Unix dell'utente (numero).

username

Il nome utente (stringa).

OpsWorks Modifiche agli agenti

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disattivato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Rilasci dell'agente Chef 12

La tabella seguente descrive importanti modifiche all'agente Chef 12 che AWS OpsWorks Stacks installa sulle istanze che gestisce.

Versione agente	Descrizione	Data di rilascio
4042	• Questa versione per agenti contiene solo modifiche minori senza nuove funzionalità	7 febbraio 2023
4041	• Questa versione per agenti contiene solo modifiche minori senza nuove funzionalità • Aggiornamento dei certificati Amazon CA	27 gennaio 2023
4040	• Questa versione per agenti contiene solo modifiche minori senza nuove funzionalità	22 luglio 2022
4039	• Correggi l'integrazione ECS per Ubuntu AMIs	30 aprile 2020
4038	• Correzione bug quando si inviano statistiche dell'istanza durante la modifica dell'ora legale	5 marzo 2020

Versione agente	Descrizione	Data di rilascio
	• Rispetto della variabile di ambiente `no_proxy` durante il download e l'installazione dell'agente	
4037	• Aggiungi il supporto per la firma delle richieste a S3 URLs senza regione utilizzando SigV4 • Rimosso il supporto per la firma delle richieste S3 utilizzando SigV2	4 giugno 2019
4035	• Correzione del bug della configurazione ECS • Correzione delle voci `fstab` duplicate dopo una modifica del tipo di istanza	8 maggio 2019
4033	• Aggiunto il supporto per Ubuntu 18.04 • Correzione del bug di installazione dell'agente in Amazon Linux 2	26 novembre 2018
4032	• Aggiunto supporto per Amazon Linux 2	24 ottobre 2018
4031	• Aggiungere supporto per Amazon Linux 2018.03 • Supporto degli archivi S3 pubblici in hosting su un altro account	15 agosto 2018
4030	• Corretta la gestione di volumi per istanze c5d	31 maggio 2018
4029	• Installa `nvme-cli` su Ubuntu 14.04 • Corretto il montaggio di volumi sulle istanze c5, m5 • Conservare sempre hostname al riavvio	2 maggio 2018
4028	• Corretta la configurazione `monit` per CentOS	20 marzo 2018
4027	• Support per il montaggio di NVMe volumi su Ubuntu 14.04 (`nvme-cli`deve essere installato manualmente) • Non richiede la proprietà `name` per i volumi	17 febbraio 2018

Versione agente	Descrizione	Data di rilascio
4026	• Monta volumi EBS NVMe basati su EBS utilizzando EBS volume ID • Corretto il montaggio di volumi EBS sulle istanze i3 • Corretto l'ordine dei volumi EBS montati su istanze c5, m5	31 gennaio 2018
4025	• Correggi la gestione dei dispositivi NVMe	13 dicembre 2017
4024	• Aggiungere supporto per Amazon Linux 2017.09	5 dicembre 2017
4023	• Aggiungi il supporto per l' CloudWatch integrazione dei log	2 Aprile 2017
4022	• Aggiornamento della versione del client Chef alla versione 12.18.31.	1 febbraio 2017
4021	• Migliorare la gestione di proxy	16 dicembre 2016
4020	• Aggiornamento della versione del client Chef alla versione 12.16.42.	8 dicembre 2016
4019	• Variabili origine proxy durante l'installazione dell'agente • Red Hat Enterprise Linux 7 ora utilizza `systemd` invece di `monit` • Non configurare EPEL su Red Hat Enterprise Linux 7 • Utilizzare `flock` invece di `lockrun.c` per elaborare il blocco • Evitare l'output strano di `ps -p1` durante il controllo per `systemd`	19 Ottobre 2016
4018	• Aggiornamento della versione del client Chef alla versione 12.13.37. • Aggiungere supporto per Amazon Linux 2016.09	25 agosto 2016

Versione agente	Descrizione	Data di rilascio
4017	• Aggiornamento della versione del client Chef alla versione 12.12.15.	10 agosto 2016
4016	• Correggere la disinstallazione dell'agente su sistemi in cui `monit` non viene utilizzato	23 giugno 2016
4015	• Correggere la configurazione ECS per Amazon Linux 2016.03	17 giugno 2016
4011	• Aggiornamento della versione del client Chef alla versione 12.10.24. • Migliorare la gestione del caricamento di log	19 maggio 2016
4008	• Aggiungere supporto per Amazon Linux 2016.03 • Aggiungere timeout all'installazione bundle • Aggiungi xfs to /etc/filesystems se esiste	16 marzo 2016
4007	• Aggiornamento della versione del client Chef alla versione 12.7.2. • Miglioramenti per la gestione degli errori per istanze locali (server ospitati al di fuori di AWS) • Migliorare la compatibilità con lo chef-sugar più recente • Riprovare il download dell'archivio per la distribuz ione	4 marzo 2016
4006	• Aggiornamento della versione del client Chef alla versione 12.6.0. • Non installare i pacchetti libxml2- devel/libxml2-dev and libxslt-devel/libxslt -dev durante l'installazione dell'agente	21 gennaio 2016
4005	• Correggere l'importazione di ec2 abilitando sempre i dati ec2 in ohai per l'infrastruttura ec2	17 dicembre 2015

Versione agente	Descrizione	Data di rilascio
4004	• AWS OpsWorks Supporto Stacks per Chef 12 Linux- Chef Client 12.5.1	3 dicembre 2015

Rilasci dell'agente Chef 11.10

La tabella seguente descrive importanti modifiche all'agente Chef 11.10 che AWS OpsWorks Stacks installa sulle istanze che gestisce.

Versione agente	Descrizione	Data di rilascio
3456	• Questa versione per agenti contiene solo modifiche minori senza nuove funzionalità • Aggiornamento dei certificati Amazon CA	27 gennaio 2023
3455	• Questa versione per agenti contiene solo modifiche minori senza nuove funzionalità	1 novembre 2022
3454	• Correggi l'integrazione ECS per Ubuntu AMIs	28 aprile 2020
3453	• Correzione bug quando si inviano statistiche dell'istanza durante la modifica dell'ora legale • Risolve il bug dei pacchetti mancanti durante la configurazione RHEL7 • Rispetto della variabile di ambiente `no_proxy` durante il download e l'installazione dell'agente	5 marzo 2020
3452	• Non includere la regione nell'URL del percorso virtuale Amazon S3 se è `us-east-1` • Estrarre e caricare libri di ricette interni nei bucket specifici della regione fase • Correggere voci `fstab` per Chef 11.10 • Rimuovere l'utilizzo SigV2 per S3 e ottenere la regione per il bucket nella richiesta	13 agosto 2019

Versione agente	Descrizione	Data di rilascio
3451	• Aggiunta del supporto per Ruby 2.6.1	20 marzo 2019
3450	• Correzione degli attributi EBS predefiniti • Installazione CloudWatchLogs dell'agente di correzione per Amazon Linux 2 • Correzione dell'installazione di bundler per rubygem versioni precedenti a 2.6.14 • Correzione del supporto per gli archivi S3 pubblici	3 dicembre 2018
3449	• Corretta la gestione di volumi per istanze c5d • Risolto il problema del supporto agli array RAID sulle NVMe istanze -device	5 giugno 2018
3448	• Aggiornamento della versione 2.3 predefinita di Ruby a 2.3.7 • Risolve il montaggio di volumi EBS su istanze NVMe basate su istanze di Ubuntu 14.04 • Supporta archivi Amazon S3 pubblici ospitati su un altro account • Corretti problemi di avvio `opsworks-agent` su istanze Red Hat Enterprise Linux	8 maggio 2018
3447	• Monta volumi EBS NVMe basati utilizzando EBS volume ID • Corretto il montaggio di volumi EBS sulle istanze i3 • Corretto l'ordine dei volumi EBS montati su c5, m5 • Aggiornamento della versione 2.3 predefinita di Ruby a 2.3.6	31 gennaio 2018
3446	• Correggi la gestione dei dispositivi NVMe • Aggiornamento della versione 2.3 predefinita di Ruby a 2.3.5	14 dicembre 2017

Versione agente	Descrizione	Data di rilascio
3445	• Aggiungere supporto per Amazon Linux 2017.09 • Aggiornamento della versione 2.2 predefinita di Ruby a 2.2.8	31 Ottobre 2017
3444	• Aggiungi il supporto per i CloudWatch registri	1 Aprile 2017
3443	• Migliorare la gestione di proxy	15 dicembre 2016
3442	• Aggiornamento della versione 2.3 predefinita di Ruby a 2.3.3 • Aggiornamento della versione 2.2 predefinita di Ruby a 2.2.6	6 dicembre 2016
3441	• Variabili origine proxy durante l'installazione dell'agente	21 Ottobre 2016
3440	• Aggiungere supporto per Amazon Linux 2016.09	13 settembre 2016
3439	• Modifiche minori, nessuna caratteristica nuova	29 luglio 2016
3438	• Aggiunto il supporto per Ruby 2.3.1 • Migliorata la registrazione di istanze con credenziali del profilo dell'istanza IAM • Rimosse rimanenze `s3curl.pl` • Correggere la configurazione ECS per Amazon Linux 2016.03	17 giugno 2016
3437	• Aggiornamento della versione 2.2 predefinita di Ruby a 2.2.5	4 maggio 2016
3436	• Aggiornamento dell'URL EPEL per Red Hat Enterprise Linux. IMPORTANTE: senza questa modifica, le istanze Red Hat Enterprise Linux non si avviano.	18 Aprile 2016

Versione agente	Descrizione	Data di rilascio
3435	• Aggiornamento della versione 2.1 predefinita di Ruby a 2.1.9 • Migliora la gestione di Amazon S3 e delle distribuzioni di archiviazione	6 Aprile 2016
3434	• Aggiungere supporto per Amazon Linux 2016.03 • Ripetizione dell'installazione di pacchetti	16 marzo 2016
3433	• Alcuni miglioramenti per le istanze locali (server ospitati all'esterno di) AWS • Migliorare la compatibilità con lo più recente`chef-sugar` • Riprovare il download dell'archivio per la distribuzione • Corretto l'URL di installazione di Ruby gems	27 febbraio 2016
3432	• Miglioramento della gestione dei caratteri speciali nei nomi di bucket • Aggiornamento di `s3_file` alla versione 2.6.6 • Salta il montaggio di volumi senza alcun punto di montaggio specificato • Riavvia sempre `unicorn` invece di arrestare e avviare per evitare tempi di inattività durante le distribuzioni • Aggiorna sempre il libro di ricette personalizzato per il comando `setup` • Dopo aver creato gli array RAID aggiorna `initramfs` per evitare problemi di mappatura del dispositivo al riavvio	20 gennaio 2016

Versione agente	Descrizione	Data di rilascio
3431	• Corretto il problema di installazione gem `passenger` e `unicorn` nel livello Rails • Aggiornamento delle versioni predefinite 2.0, 2.1 e 2.2 di Ruby a 2.0.0p648, 2.1.8 e 2.2.4 • Consenti nomi di pacchetti `postgres` da impostare in un JSON personalizzato • Aggiornamento della versione predefinita di Node.js a 0.12.9	22 dicembre 2015
3430	• Modifiche minori, nessuna caratteristica nuova	25 Novembre 2015
3429	• Migliora la demonizzazione degli OpsWorks agenti (chiudi stdout/stderr) • Miglioramento della solidità della risorsa `s3_file` (nuovi tentativi, eccezioni rilevate)	18 Novembre 2015
3428	• Aggiungendo `postgres` il rilevamento dell'adat tatore basato sul Gemfile, corregge opsworks-cookbooks/issues/136 https://github.com/aws/	17 giugno 2016
3427	• Corretto un problema con il recupero delle credenziali nell'agente • Aggiornamento delle versioni predefinite 2.0, 2.1 e 2.2 di Ruby a 2.0.0p647, 2.1.7 e 2.2.3	11 settembre 2015

Versione agente	Descrizione	Data di rilascio
3426	• Aggiornato `aws-sdk` a 1.65.0 • Miglioramento del download da Amazon S3 sostituendolo con `s3curl` `s3_file cookbook` • Modifica della versione predefinita di Node.js a 0.12.7 • Registrazione di log aggiunta per le app Node.js. STDOUT e STDERR registrati e ruotati nella directory `shared/log` • Aggiornamento del checkout del modulo secondario del libro di ricette personalizzato esplicito • È stata aggiunta una soluzione alternativa per https://github.com/aws/ opsworks-cookbooks/ issues/213 che verificherà che i bind mount siano stati eseguiti prima della creazione della directory `deploy`	27 agosto 2015
3425	• Supporto ECS per Amazon Linux e Ubuntu	27 luglio 2015
3424	• Modifiche minori, nessuna caratteristica nuova	9 luglio 2015
3422	• Supporto completo di Red Hat Enterprise Linux 7 • Generazione di `/etc/hosts` più resiliente agli errori	29 giugno 2015
3421	• Opzione per sovrascrivere il nome del pacchetto di database per Red Hat Enterprise Linux 7 • Aggiornamento della configurazione `monit` `systemd` per evitare che `systemd` invii il segnale `kill` ai processi monitorati da `monit`	11 giugno 2015

AWS OpsWorks Risorse Stacks

> **Important**
>
> Il AWS OpsWorks Stacks servizio ha raggiunto la fine del ciclo di vita il 26 maggio 2024 ed è stato disabilitato sia per i clienti nuovi che per quelli esistenti. Consigliamo vivamente ai clienti di migrare i propri carichi di lavoro verso altre soluzioni il prima possibile. Se hai domande sulla migrazione, contatta il Supporto AWS Team su AWS re:post o tramite Premium AWS Support.

Le seguenti risorse correlate possono rivelarsi utili durante l'utilizzo di questo servizio.

Guide di riferimento, strumenti e risorse di supporto

Utili guide, forum, informazioni di contatto e altre risorse sono disponibili tramite AWS OpsWorks Stacks e Amazon Web Services.

- AWS OpsWorks Riferimento all'API Stacks: descrizioni, sintassi ed esempi di utilizzo delle azioni e dei tipi di dati di AWS OpsWorks Stacks, inclusi parametri e codici di errore comuni.
- AWS OpsWorks Domande frequenti tecniche su Stacks: le domande più frequenti poste dagli sviluppatori su questo prodotto.
- AWS OpsWorks Stacks Release Notes: una panoramica di alto livello della versione corrente. In particolare, in questo documento vengono indicate tutte le nuove funzioni, le correzioni e tutti i problemi noti.
- AWS Tools for PowerShell: un set di PowerShell cmdlet Windows che espongono le funzionalità dell'ambiente AWS SDK per .NET . PowerShell
- Interfaccia a riga di comando AWS: una sintassi a riga di comando uniforme per l'accesso ai servizi AWS. L'interfaccia a riga di comando di AWS utilizza un singolo processo di configurazione per abilitare l'accesso per tutti i servizi supportati.
- AWS OpsWorks Riferimento alla riga di comando di Stacks: comandi AWS OpsWorks specifici per Stacks da utilizzare al prompt della riga di comando.

- Corsi e workshop: collegamenti a corsi specializzati e basati su ruoli, oltre a laboratori di autoapprendimento per aiutarti ad affinare le tue abilità e acquisire esperienza pratica. AWS

- AWS Developer Center: esplora i tutorial, scarica strumenti e scopri gli eventi per sviluppatori. AWS
- AWS Strumenti per sviluppatori: collegamenti a strumenti di sviluppo SDKs, toolkit IDE e strumenti da riga di comando per lo sviluppo e la gestione di applicazioni. AWS
- Centro risorse introduttivo: scopri come configurare Account AWS, entrare a far parte della AWS community e lanciare la tua prima applicazione.
- Tutorial pratici: segui i tutorial per avviare la step-by-step tua prima applicazione su. AWS
- AWS Whitepaper: collegamenti a un elenco completo di AWS white paper tecnici, su argomenti quali architettura, sicurezza ed economia e redatti da Solutions Architects o altri esperti tecnici. AWS
- Supporto AWS Center: l'hub per la creazione e la gestione dei casi. Supporto AWS Include anche collegamenti ad altre risorse utili, come forum, informazioni tecniche FAQs, stato di salute del servizio e AWS Trusted Advisor.
- Supporto— La pagina web principale per informazioni su Supporto one-on-one, un canale di supporto a risposta rapida per aiutarti a creare ed eseguire applicazioni nel cloud.
- Contatti - Un punto di contatto centrale per richieste relative a fatturazione, account, eventi, uso illecito e altre questioni relative ad AWS .
- AWS Termini del sito: informazioni dettagliate sul nostro copyright e marchio, sull'account, sulla licenza e sull'accesso al sito e altri argomenti.

AWS Kit di sviluppo software

Amazon Web Services fornisce kit di sviluppo software per accedere agli AWS OpsWorks stack da diversi linguaggi di programmazione. Le librerie SDK automatizzano varie attività comuni, tra cui la firma crittografica delle richieste di servizio, i nuovi tentativi di richieste o la gestione delle risposte errate.

- AWS SDK per Java— Configurazione e altra documentazione
- AWS SDK per .NET— Configurazione e altra documentazione.
- SDK AWS per PHP — Documentazione
- AWS SDK per Ruby— Documentazione
- altra documentazione
- AWS SDK for Python (Boto)— Configurazione e altra documentazione

Software open source

AWS OpsWorks Stacks include una varietà di pacchetti software open source, regolati dalle rispettive licenze. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per le istanze Chef 12 Linux, apri il file `THIRD_PARTY_LICENSES` nella directory `/opt/aws/opsworks/current` sull'istanza.
- Per Chef 11.10 e versioni precedenti per Linux, scarica il OpsWorks Linux Agent Attributions Document PDF.

AWS OpsWorks cronologia dei documenti

Modifica	Descrizione	Data
Aggiornamenti agli stacks AWS OpsWorks	Ora puoi usare lo strumento Detach in Place per scollegare le tue OpsWorks istanze dal servizio OpsWorks Stacks, vedi Uso dello strumento AWS OpsWorks Stacks Detach in Place in questa guida.	11 aprile 2024
Aggiornamenti a Stacks AWS OpsWorks	Ora puoi migrare i tuoi AWS OpsWorks Stacks ad AWS Systems Manager Applicati on Manager utilizzando uno script di migrazione. Per ulteriori informazioni, consulta Migrazione delle applicazi oni ad AWS OpsWorks StacksAWS Systems Manager Application Manager in questa guida.	22 dicembre 2022
Aggiornamenti a e AWS OpsWorks for Chef AutomateA WS OpsWorks for Puppet Enterprise	È ora disponibile una procedura di risoluzione dei problemi che descrive cosa fare se la manutenzione del sistema fallisce per il tuo AWS OpsWorks for Chef Automate o OpsWorks per il server Puppet Enterprise. Per ulteriori informazioni, consulta Problemi di manutenzione del sistema per il server Chef Automate o Errori di	29 settembre 2022

	manutenzione del sistema per il server Puppet Enterprise. in questa guida.	
Aggiornamenti a AWS OpsWorks for Chef Automate e AWS OpsWorks for Puppet Enterprise	Una procedura di risoluzione dei problemi è ora disponibile se il tuo server AWS OpsWorks for Chef Automate o OpsWorks per Puppet Enterprise entra in uno `Connection lost` stato. Per ulteriori informazioni, vedi Il server Chef Automate è in uno `Connection lost` stato o Il server Puppet Enterprise è in uno `Connection lost` stato in questa guida.	23 marzo 2022
Aggiornamenti a Stacks AWS OpsWorks	Come best practice in materia di sicurezza, ora puoi aggiungere una chiave o (`aws:SourceArn` o `aws:SourceAccount` entrambe) alle politiche di relazione fiduciaria che consentono a Stacks di accedere a AWS OpsWorks Stacks per eseguire attività in altri AWS servizi. Per ulteriori informazioni, consulta la sezione Prevenzione intersettoriale degli agenti confusi in AWS OpsWorks Stacks in questa guida.	4 marzo 2022

Aggiornamenti a e AWS OpsWorks for Chef AutomateAWS OpsWorks for Puppet Enterprise	Come best practice in materia di sicurezza, ora puoi aggiungere una chiave di `aws:SourceAccount` condizione `aws:SourceArn` o (o entrambe) alle politiche di relazione fiduciaria che consentono AWS OpsWorks for Chef Automate l'accesso a Puppet Enterprise OpsWorks per eseguire attività in altri AWS servizi. Per ulteriori informazioni, consulta la sezione Prevenzione interservizio confusa tra diversi servizi in questa guida.	10 gennaio 2022
Aggiornamenti a e AWS OpsWorks for Chef AutomateAWS OpsWorks for Puppet Enterprise	AWS OpsWorks for Chef Automate e OpsWorks per Puppet Enterprise hanno aggiornato le politiche gestite `AWSOpsWorksCMServiceRole` e `AWSOpsWorksCMInstanceProfileRole` ora archiviano i segreti in AWS Secrets Manager.	3 maggio 2021

Aggiornamenti a AWS OpsWorks for Puppet Enterprise	La versione del motore di un server OpsWorks for Puppet Enterprise che crei nella console è ora la 2019.8.5. Utilizzando l'API, puoi specificare una delle due versioni `2019` o `2017` quando crei un server Puppet Enterprise. L'`DescribeServers` API ora restituisce un attributo chiamato `PUPPET_API_CRL` nei risultati. Questo attributo contiene un elenco di revoca dei certificati per uso interno.	28 Aprile 2021
AWS OpsWorks Stacks utilizza una nuova politica gestita	AWS OpsWorks Stacks ha modificato la politica gestita che include le autorizzazioni per eseguire tutte le azioni in Stacks. AWS OpsWorks La nuova politica è. AWSOpsWorks_FullAccess Per ulteriori informazioni sulle autorizzazioni contenute in questa politica, consulta Criteri di esempio.	19 febbraio 2021
Migrazione degli AWS OpsWorks Stacks stack da EC2 -Classic a un VPC	È stata aggiunta la documentazione che descrive come migrare uno AWS OpsWorks Stacks stack da EC2 -Classic a un VPC.	29 settembre 2020

Rigenera uno starter kit per e AWS OpsWorks for Chef AutomateAWS OpsWorks for Puppet Enterprise	È stata aggiunta la documentazione che descrive come rigenerare lo starter kit per uno o un server. AWS OpsWorks for Chef Automate AWS OpsWorks for Puppet Enterprise	29 luglio 2020
AWS OpsWorks for Puppet Enterprise consente di creare un server che utilizza un dominio, un certificato e una chiave privata personalizzati	Ora puoi creare un server OpsWorks for Puppet Enterprise che utilizza un dominio, un certificato e una chiave privata personalizzati. È possibile aggiornare un server Chef Automate 2.0 esistente per utilizzare un dominio personalizzato creando un server da un backup di un server esistente.	17 aprile 2020
AWS OpsWorks for Chef Automate e AWS OpsWorks for Puppet Enterprise ora supporta l'etichettatura nella console	È ora possibile aggiungere tag a un AWS OpsWorks for Chef Automate server o a un AWS OpsWorks for Puppet Enterprise master o ai backup del server, utilizzando il AWS Management Console o il. AWS CLI Per ulteriori informazioni, vedere Lavora con i tag (Chef) o Lavora con i tag (Puppet).	26 febbraio 2020

AWS OpsWorks for Chef Automate semplifica l'aggiornamento dei server Chef Automate 1 esistenti a Chef Automate 2	Puoi aggiornare AWS OpsWorks for Chef Automate i server idonei che eseguono Chef Automate 1 a Chef Automate 2 scegliendo Avvia aggiornamento nella pagina dei dettagli del server nella console o eseguendo l'azione API. `StartMaintenance` Per ulteriori informazioni, consulta Aggiornare un AWS OpsWorks for Chef Automate server a Chef Automate 2.	24 gennaio 2020
AWS OpsWorks for Chef Automate e AWS OpsWorks for Puppet Enterprise	Un nuovo capitolo sulla sicurezza in AWS OpsWorks CM (AWS OpsWorks for Chef Automate and AWS OpsWorks for Puppet Enterprise) è stato aggiunto alla guida.	23 dicembre 2019
AWS OpsWorks for Chef Automate e AWS OpsWorks for Puppet Enterprise supporta il tagging	È ora possibile aggiungere tag a un AWS OpsWorks for Chef Automate server o a un AWS OpsWorks for Puppet Enterprise master o ai backup del server utilizzando. AWS CLI AWS OpsWorks CM ora supporta l'autorizzazione basata su tag.	18 dicembre 2019

AWS OpsWorks for Chef Automate consente di creare un server che utilizza un dominio, un certificato e una chiave privata personalizzati	È ora possibile creare un server AWS OpsWorks for Chef Automate 2.0 che utilizza un dominio, un certificato e una chiave privata personalizzati. È possibile aggiornare un server Chef Automate 2.0 esistente per utilizzare un dominio personalizzato creando un server da un backup di un server esistente.	22 ottobre 2019
AWS OpsWorks Stacks ora supporta Ruby 2.6.1	AWS OpsWorks Stacks supporta Ruby 2.6.1 sui livelli Rails App Server negli stack Chef 11.10.	2 maggio 2019
AWS OpsWorks for Chef Automate ora supporta Chef Automate 2.0	AWS OpsWorks for Chef Automate I nuovi server eseguiranno Chef Automate 2.0, che include aggiornamenti a Chef InSpec, nuove funzionalità di scansione e reportistica di conformità e Chef Infra.	30 aprile 2019
AWS OpsWorks for Chef Automate e AWS OpsWorks for Puppet Enterprise	Ora puoi utilizzarlo AWS CloudFormation per creare un AWS OpsWorks for Chef Automate server o un server AWS OpsWorks for Puppet Enterprise master.	24 gennaio 2019
AWS OpsWorks Stack	AWS OpsWorks Stacks ora supporta le istanze che eseguono Ubuntu 18.04 LTS negli stack Chef 12.	18 dicembre 2018

AWS OpsWorks per Puppet Enterprise	È stata aggiunta una procedura per configurare una connessione basata su SSH a un repository di controllo che utilizza. CodeCommit	3 dicembre 2018
AWS OpsWorks Stack	AWS OpsWorks Stacks ora supporta le istanze che eseguono Amazon Linux 2 negli stack Chef 12.	15 novembre 2018
AWS OpsWorks Stack	AWS OpsWorks Stacks ora supporta le istanze che eseguono Amazon Linux 2018.03 negli stack Chef 11.10.	23 ottobre 2018
AWS OpsWorks Stack	AWS OpsWorks Stacks ora supporta le istanze che eseguono Amazon Linux 2018.03 negli stack Chef 12.	23 agosto 2018
AWS OpsWorks for Chef Automate OpsWorks e per Puppet Enterprise	OpsWorks for Puppet Enterprise è stato aggiornato a PE 2018.1.2. AWS OpsWorks for Chef Automate è stato aggiornato a Chef Automate 1.8.68.	29 giugno 2018

- AWS OpsWorks for Chef Automate e OpsWorks per la versione dell'API Puppet Enterprise: 2016-11-01
- AWS OpsWorks Versione dell'API Stacks: 2016-03-08
- Ultimo aggiornamento della documentazione: 2024-04-11

Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti apportate a ogni versione della Guida per l'utente di AWS OpsWorks prima di giugno 2018.

Descrizione	Data
AWS OpsWorks La versione di Stacks Chef per gli stack basati su Windows è stata aggiornata alla 12.22; la versione di Ruby è ora 2.3.6.	19 aprile 2018
Nuove procedure per creare un AWS OpsWorks for Chef Automate server o un master for Puppet Enterprise utilizzando il. OpsWorks AWS CLI	23 marzo 2018
Versione Chef Automate aggiornata alla 1.8; La configurazione di Chef Compliance è stata semplificata con l'aggiunta del ricettario. `opsworks-audit`	5 marzo 2018
È stato aggiunto il supporto per gli eventi AWS OpsWorks Stacks in Amazon CloudWatch Events.	20 febbraio 2018
È stato aggiunto il supporto per nuovi tipi di volume EBS in AWS OpsWorks Stacks e una nuova API,. DescribeOperatingSystems	25 gennaio 2018
OpsWorks per Puppet Enterprise e AWS OpsWorks for Chef Automate ora supporta la selezione di più gruppi di sicurezza durante la creazione di un server.	18 gennaio 2018
È stato aggiunto il supporto per AWS OpsWorks Stacks nella regione Europa (Parigi).	19 dicembre 2017
È stato aggiunto il supporto per AWS OpsWorks for Chef Automate e OpsWorks per Puppet Enterprise in sei regioni aggiuntive e sono state aggiunte procedure per la creazione di backup di AWS OpsWorks for Chef Automate e OpsWorks per i server Puppet Enterprise nella. AWS Management Console	18 dicembre 2017
Aggiunto il nuovo servizio e la OpsWorks documentazione di Puppet Enterprise.	16 novembre 2017

Descrizione	Data
È stato aggiunto il supporto per Amazon Linux 2017.09 a AWS OpsWorks Stacks.	7 novembre 2017
È stato aggiunto il supporto per Chef Compliance a. AWS OpsWorks for Chef Automate	25 ottobre 2017
È stato aggiunto il supporto per Amazon Linux 2017.09 a. AWS OpsWorks for Chef Automate	9 Ottobre 2017
È stato aggiunto l'argomento relativo alla manutenzione del sistema al capitolo. AWS OpsWorks for Chef Automate	28 luglio 2017
È stato aggiunto il supporto per i tag in AWS OpsWorks Stacks.	6 giugno 2017
Aggiunta integrazione con CloudWatch Logs.	10 aprile 2017
Aggiunto il nuovo AWS OpsWorks for Chef Automate servizio e la nuova documentazione.	1° dicembre 2016
È stato aggiunto il supporto per l'endpoint regionale degli Stati Uniti orientali (Ohio).	12 Ottobre 2016
Aggiunto il supporto per gli stack e le istanze che eseguono il sistema operativo Amazon Linux 2016.09.	30 settembre 2016
È stato aggiunto il supporto per la regione Asia Pacifico (Seoul) e nove endpoint regionali aggiuntivi.	15 agosto 2016
Aggiunto il supporto per Node.js 0.12.15 e Ruby 2.3 in livelli integrati.	6 luglio 2016
È stato aggiunto il supporto per la regione Asia Pacifico (Mumbai).	28 giugno 2016
Aggiunto il supporto per gli stack e le istanze che eseguono il sistema operativo CentOS 7.	22 giugno 2016
È stata aggiunta una descrizione CodePipeline dettagliata e l'integra zione con Stacks. AWS OpsWorks	2 giugno 2016

Descrizione	Data
Aggiunto il supporto per gli stack e le istanze che eseguono il sistema operativo Ubuntu 16.04 LTS.	1 giugno 2016
Aggiunto il supporto per Chef 12 Linux e la relativa documentazione.	3 dicembre 2015
Aggiunta la procedura guidata Node.js alle Nozioni di base.	14 luglio 2015
Aggiunti due nuovi esempi di libro di ricette a Cookbooks 101.	14 luglio 2015
Aggiunto il supporto per la gestione della versione agente.	23 giugno 2015
Aggiunto il supporto per gestire la versione agente.	24 giugno 2015
È stato aggiunto il supporto per Windows personalizzato. AMIs	22 giugno 2015
Aggiunti tre nuovi argomenti di Best practice.	11 giugno 2015
Aggiunto il supporto per gli stack di Windows.	18 maggio 2015
Aggiunto un capitolo sulle Best practice.	15 dicembre 2014
È stato aggiunto il supporto per il drenaggio della connessione Elastic Load Balancing e i timeout di spegnimento personalizzati.	15 dicembre 2014
È stato aggiunto il supporto per la registrazione di istanze create al di fuori di OpsWorks AWS Stacks.	9 dicembre 2014
È stato aggiunto il supporto per Amazon SWF.	4 settembre 2014
Aggiunto il supporto per l'associazione di variabili di ambiente con le applicazioni e Cookbooks 101 esteso.	16 luglio 2014
Aggiunto Cookbooks 101, un'introduzione tutorial per l'impleme ntazione dei libri di ricette.	16 luglio 2014
È stato aggiunto il supporto per CloudTrail.	4 giugno 2014
È stato aggiunto il supporto per Amazon RDS.	14 maggio 2014

Descrizione	Data
Aggiunto il supporto per Chef 11.10 e Berkshelf.	27 marzo 2014
È stato aggiunto il supporto per i volumi Amazon EBS PIOPS.	16 dicembre 2013
Aggiunte le autorizzazioni basate su risorsa.	5 dicembre 2013
Aggiunta la gestione delle risorse.	7 ottobre 2013
È stato aggiunto il supporto per. VPCs	29 agosto 2013
Aggiunto il supporto per custom AMIs e Chef 11.4.	24 luglio 2013
Aggiunto il supporto della console per più livelli per istanza.	1 luglio 2013
È stato aggiunto il supporto per le istanze supportate da Amazon EBS, Elastic Load Balancing e il monitoraggio di Amazon. CloudWatch	14 maggio 2013
Versione iniziale della AWS OpsWorks Stacks User Guide.	18 febbraio 2013

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.